NAVAL POSTGRADUATE SCHOOL
MONTEREY, CALIFORNIA

THESIS

TOWARD ASSESSMENT OF DOMINANT BATTLESPACE AWARENESS: A REMOTE SENSOR SYSTEM MODEL

by

Kenneth H. Munson, Jr.

March, 1996

Thesis Advisor: Donald P. Gaver
Second Reader: Arnold H. Buss

REPORT DOCUMENTATION PAGE

Form Approved OMB No. 0704-0188

1. AGENCY USE ONLY *(Leave blank)*	2. REPORT DATE March, 1996	3. REPORT TYPE AND DATES COVERED Master's Thesis

4. TITLE AND SUBTITLE Toward Assessment of Dominant Battlespace Awareness: A Remote Sensor System Model	5. FUNDING NUMBERS
6. AUTHOR(S) Kenneth H. Munson, Jr.	
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Naval Postgraduate School Monterey CA 93943-5000	8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

12a. DISTRIBUTION/AVAILABILITY STATEMENT Approved for public release; distribution is unlimited.	12b. DISTRIBUTION CODE

13. ABSTRACT *(maximum 200 words)*

Two broad concepts have begun to permeate U.S. military strategic planning since the end of the Gulf War: the revolution in military affairs (RMA) and dominant battlespace awareness (DBA). An RMA represents a basic change in the conduct of warfare which incorporates new technologies, operational innovation and organizational changes. DBA refers to the military's ability to efficiently obtain and effectively use information to dominate an opposing force. This thesis is a study of a stylized warfare scenario involving elements of DBA and RMA. Specifically, U.S. attack aircraft attempt to prevent enemy transporter-erector-launchers (TELs) from harassing neighboring countries with theater ballistic missiles. The U.S. aircraft may be aided by use of unattended ground sensors (UGSs); the enemy TEL activities are correspondingly enhanced by decoy TELs. The model described allows the combat advantage of each side to be quantitatively compared. Trend analysis demonstrates the benefits of deception and the potential of UGSs.

14. SUBJECT TERMS Unattended Ground Sensors, Dominant Battlespace Awareness, Revolution in Military Affairs, Theater Ballistic Missiles, Attack Operations.	15. NUMBER OF PAGES 112
	16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT Unclassified	18. SECURITY CLASSIFICATION OF THIS PAGE Unclassified	19. SECURITY CLASSIFICATION OF ABSTRACT Unclassified	20. LIMITATION OF ABSTRACT UL

NSN 7540-01-280-5500

Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std. 239-18 298-102

TOWARD ASSESSMENT OF DOMINANT BATTLESPACE AWARENESS: A REMOTE SENSOR SYSTEM MODEL

Kenneth H. Munson, Jr.
Lieutenant, United States Navy
B.S., Pennsylvania State University, 1989

Submitted in partial fulfillment
of the requirements for the degree of

MASTER OF SCIENCE IN OPERATIONS RESEARCH

from the

NAVAL POSTGRADUATE SCHOOL
MARCH 1996

Author: Kenneth H Munson

Kenneth H. Munson, Jr.

Approved by: Donald P. Gaver

Donald P. Gaver, Thesis Advisor

Arnold H. Buss

Arnold H. Buss, Second Reader

Frank C. Petho

Frank Petho, Chairman
Department of Operations Research

ABSTRACT

Two broad concepts have begun to permeate U.S. military strategic planning since the end of the Gulf War: the revolution in military affairs (RMA) and dominant battlespace awareness (DBA). An RMA represents a basic change in the conduct of warfare which incorporates new technologies, operational innovation and organizational changes. DBA refers to the military's ability to efficiently obtain and effectively use information to dominate an opposing force. This thesis is a study of a stylized warfare scenario involving elements of DBA and RMA. Specifically, U.S. attack aircraft attempt to prevent enemy transporter-erector-launchers (TELs) from harassing neighboring countries with theater ballistic missiles. The U.S. aircraft may be aided by use of unattended ground sensors (UGSs); the enemy TEL activities are correspondingly enhanced by decoy TELs. The model described allows the combat advantage of each side to be quantitatively compared. Trend analysis demonstrates the benefits of deception and the potential of UGSs.

THESIS DISCLAIMER

The reader is cautioned that computer programs developed in this research may not have been exercised for all cases of interest. While every effort has been made, within the time available, to ensure that the programs are free of computational and logic errors, they cannot be considered validated. Any application of these programs without verification is at the risk of the user.

TABLE OF CONTENTS

I. INTRODUCTION 1

A. BACKGROUND 1

B. THESIS SCOPE AND OBJECTIVES 2

II. SHAPING THE REVOLUTION 3

A. A REVOLUTION IN MILITARY AFFAIRS 3

B. THE TECHNOLOGY OF THE REVOLUTION 5

1. Defining the Threat 6

2. Unattended Ground Sensors (UGSs) 7

III. SIMULATION DEVELOPMENT 11

A. WARFARE SCENARIO 11

B. MODEL ASSUMPTIONS 11

C. DESIGN CONSIDERATIONS 13

1. Attack Aircraft 13

2. UGS Network 15

3. Transporter-Erector-Launcher Activity Cycle 17

4. TEL Decoys 17

D. INPUT PARAMETERS 19

IV. DISCUSSION OF RESULTS 21

A. ANALYTICAL MODELS 22

1. The Independent Markovian Model 22

2. The Wave Model 27

B. COMPARISON OF SIMULATION AND ANALYTICAL MODELS . . 31

C. CASE COMPARISON 40

1. Case 1 versus Case 2 41

2. Case 3 versus Case 4 44

3. Case 1 versus Case 3 46

4. Case 2 Versus Case 4 47

5. Inclusive Comparison 47

V. CONCLUSION 51

A. SUMMARY 51

B. ADDITIONAL RESEARCH RECOMMENDATIONS 53

APPENDIX A. PROLIFERATION OF TBMS 55

APPENDIX B. COMPARISON CASE RESULTS 57

APPENDIX C. SIMULATION PROGRAM 61

LIST OF REFERENCES 93

INITIAL DISTRIBUTION LIST 95

EXECUTIVE SUMMARY

Two broad concepts have begun to permeate U.S. military strategic planning since the end of the Gulf War: the revolution in military affairs (RMA) and dominant battlespace awareness (DBA). An RMA represents a basic change in the conduct of warfare which incorporates new technologies, operational innovation and organizational changes. Admiral William A. Owens, the former Vice Chairman of the Joint Chiefs of Staff has forcefully called attention to the nature of the current RMA, and the challenges and opportunities it presents to the U.S. His framework for the design of U.S. forces centered around three concepts: battlespace awareness, advanced C^4I, and precision force use. The intersection of battlespace awareness and advanced C^4I yields the term dominant battlespace awareness (DBA). DBA refers to the military's ability to efficiently obtain and effectively use information to dominate an opposing force.

This thesis is a study of a stylized warfare scenario involving elements of DBA and RMA. Specifically, U.S. attack aircraft attempt to prevent enemy transporter-erector-launchers (TELs) from harassing neighboring countries with theater ballistic missiles. The U.S. aircraft may be aided by use of unattended ground sensors (UGSs); the enemy TEL activities are correspondingly enhanced by decoy TELs. A computer simulation is developed in which U.S. forces operate with limited DBA and enemy forces experience the first stages of an RMA.

Two simplified and computationally tractable analytical models based upon the methodology of the simulation are formulated to supplement the simulation. Numerical results from the analytical models and the simulation are contrasted and compared to gain insight into expected behavior of the simulation. The analytical models help to explain the simulation output and provide numerical bounds on simulation results for the input parameters used in the comparison.

Results are generated from input parameters for four comparison cases. First the model is run with aircraft searching for TELs. The second case adds decoy TELs that remain visible throughout the simulation in an attempt to absorb attacks from the aircraft. In the third case the decoy TELs are not present and the aircraft search for TELs with the assistance of a UGS network. All four components are included (TELs, decoy TELs, aircraft and UGS) in the final comparison case.

Examination of the output from the analytical model or the simulation is in terms of two measures of effectiveness (MOEs). The first is based on the length of time until the aggressor nation chooses to cease hostilities. In this model the TELs continue to conduct attacks until all TELs have been destroyed. The time until the last TEL is destroyed is called the *survival time*. The first MOE provided will be an estimator of the expected survival time. The second MOE is based on the total number of theater ballistic missiles launched by the TELs during hostilities. The simulation determines an estimator of the expected total number of launches.

The results of any simulation are dependent on the simulation itself and thus are not presented as predictions but used to highlight important trends. There are two important trends demonstrated from the output of the simulation. First, the importance of deception in warfare (especially to those seeking to deceive U.S. forces) is reinforced. Second, the unattended ground sensor system modeled in the simulation provides an enhanced capability to conduct attack operations lending credence to the development of such a system.

I. INTRODUCTION

A. BACKGROUND

Two broad concepts have begun to permeate U.S. military strategic planning since the end of the Gulf War: the revolution in military affairs (RMA) and dominant battlespace awareness (DBA). An RMA represents a basic change in the conduct of warfare which incorporates new technologies, operational innovation and organizational changes. Admiral William A. Owens, the former Vice Chairman of the Joint Chiefs of Staff has forcefully called attention to the nature of the current RMA, and the challenges and opportunities it presents to the U.S. His framework for the design of U.S. forces centered around three concepts: battlespace awareness, advanced C^4I, and precision force use. The intersection of battlespace awareness and advanced C^4I yields the term dominant battlespace awareness (DBA). DBA refers to the military's ability to efficiently obtain and effectively use information to dominate an opposing force.

When considering this revolution, it is important to note that not only United States forces are permitted to experience the technological change associated with an RMA. A revolution in military affairs may occur in any nation. It has been proposed that the United States will experience this revolution sooner than the rest of the world (Owens, 1995). While most agree that the United States has the best opportunity to promote a new RMA, other nations will seek to enhance their military effectiveness to the best of their abilities. U.S. military thinkers should thus consider other nations' ability to develop revolutionary capabilities.

Post-Gulf-War analysis indicates the threat of mobile theater ballistic missiles (TBMs) is still substantial. Although coalition forces in the Gulf War had air superiority, the sorties flown with TBM missions did not prove to be an effective system for locating the transporter-erector-launchers (TELs) that serve as the mobile platform for TBMs (Cohen, 1993). As nations promote the "TBM revolution," the U.S. should devote increased attention to countering TELs and other such weapon systems. Much of the initial countering effort of the Gulf War focused on post-launch interception of TBMs. As demonstrated by Ehlers (1992), a reduction in TBM effectiveness can be achieved by focusing on destroying the launch platform. Actions to neutralize an adversary's TBM capability are known as attack operations. There have been several proposals for a system of unattended ground sensors (UGSs) to detect enemy TELs (Berhow, 1993 and Junker, 1995). Developed during the Vietnam War, UGS technology matured. Under the premise that U.S. forces will operate with DBA, a system of UGSs that trigger attack aircraft would seem to be useful for combating the threat of mobile TBMs.

B. THESIS SCOPE AND OBJECTIVES

This thesis is a study of a stylized warfare scenario involving elements of DBA and RMA. Specifically, U.S. attack aircraft attempt to prevent enemy transporter-erector-launchers (TELs) from harassing neighboring countries with theater ballistic missiles. The U.S. aircraft may be aided by use of unattended ground sensors (UGSs); the enemy TEL activities are correspondingly enhanced by decoy TELs. A computer simulation is developed in which U.S. forces operate with limited DBA and enemy forces experience the first stages of an RMA.

II. SHAPING THE REVOLUTION

A. A REVOLUTION IN MILITARY AFFAIRS

A nation's realization that a revolution in military affairs has occurred often is only achieved in hindsight. The proponents of the current RMA discussion theorize that, with the foresight of the coming revolution, the United States can place itself on the cutting edge of the revolution's promotion. The standard indications that an RMA is occurring are advances in the technology of warfare. Technological development alone does not provide a guarantee of victory in war. Changes in the way a military is organized and conducts operations with this technology allow that nation to participate in the revolution.

There are three preconditions to the emergence of an RMA (Fitzsimonds, 1994):

- *Technological Development* - Intuitively the most obvious requirement to initiate an RMA. Technological development has grown exponentially since the dawn of the Industrial Revolution. New military technology does not come fully integrated into a military system.

- *Operational Innovation* - How to best employ the potential of any military system is never immediately known. Military organizations devote considerable energy toward devising doctrine for new technologies to avoid trial by fire.

- *Organizational Adaptation* - Given the static nature of any bureaucracy, a nation's military must experience a cultural change when incorporating new technologies and operational capabilities.

The three primary preconditions listed above must occur in concert for an RMA to truly take place. Additionally, the revolution need not occur during actual hostilities and in fact generally occurs during periods of peace (Fitzsimonds, 1994). With the Persian Gulf War fading farther into history, the United States seems to have entered just such a period.

If the United States has truly entered its own RMA, what form will U.S. forces take to meet the challenges of this revolution? A prominent architect of a framework for the discussion has been Admiral William A. Owens, former Vice Chairman of the Joint Chiefs of Staff. He has advocated shaping future U.S. forces in terms of desired capabilities rather than threat. As a proponent of the idea that the U.S. is currently experiencing a revolution in military affairs, he has pointed to three areas of emerging technology (Owens, 1995):

- Intelligence, surveillance, and reconnaissance (ISR)
- Advanced command, control, communications, computers, and intelligence (Advanced C^4I)
- Precision-guided munitions (PGMs)

Continuing to define his vision, Admiral Owens viewed the technologies mentioned above as a new "System of Systems." The system architecture comprises three primary concepts (Figure 1.):

- *Battlespace Awareness* involves the technologies associated with ISR. These systems are responsible for providing tactical commanders with a picture of the battlefield. Also included is information concerning weather, terrain, and electromagnetic characteristics of the area.
- *Advanced C^4I systems* are those that handle data and information transfer. This information is critical to a commander for it provides increased awareness of enemy intentions and actions. Friendly forces also benefit as increased understanding leads to efficient target identification and assignment.
- *Precision Force Use* is primarily concerned with the use of precision guided weapon technology. As military thinkers have promoted the notion of an RMA, precision force use has also come to include our ability to coordinate the first two concepts into precise offensive action.

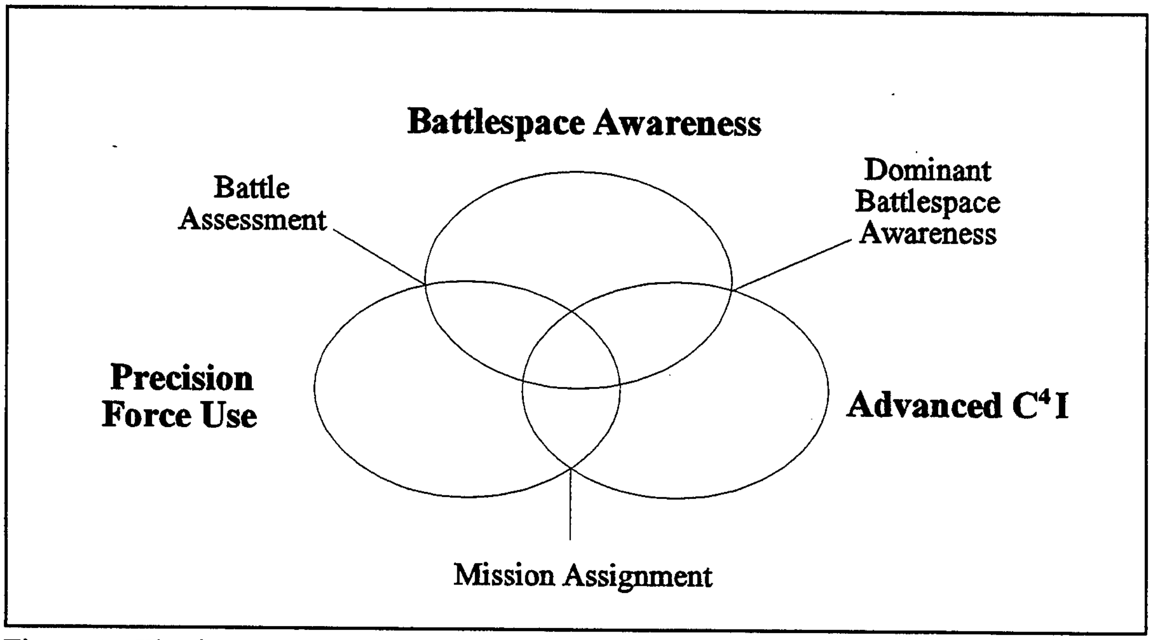

Figure 1. The framework for discussion of United States participation in a revolution in military affairs (Owens, 1995).

At the intersection of each of these core concepts are three secondary concepts. Drawn from the definitions provided previously each can be summarized by the following:

- *Mission Assignment* - refers to the ability to assign targets to weapon systems. Often called "Good Targeting," it calls for an effective link of sensor to shooter.

- *Battle Assessment* - represents an ability to accurately determine the outcome of any attack. This has been called "Good Bomb Damage Assessment."

- *Dominant Battlespace Awareness* (DBA) - The capability to achieve real-time, all weather, continuous surveillance in and over a large geographical area that is sufficient to determine the presence, movement, and state of objects, emissions, activities or events of military interest. Simply said, DBA represents "Good Knowledge."

B. THE TECHNOLOGY OF THE REVOLUTION

As stated previously this thesis is concerned with examining an aspect of the revolution in military affairs as it affects both the United States and any potential adversary.

Although there are ample areas for study, this thesis focuses on two technologies. The first is best described as a threat technology. Ballistic missiles are the threat of concern, specifically mobile theater ballistic missiles. The second technology of interest is part of attack operations. The focus here is on the use of unattended ground sensors in a real-time cueing network for overhead attack aircraft.

1. Defining the Threat

When providing the framework for the presumed threat technology, one must begin with a justification establishing the importance of the technologies considered. This section begins with a brief discussion of missile proliferation. Then the current threat is depicted with the goal of establishing the basis for a TBM revolution.

Since the end of the Gulf War, U.S. military strategists have conducted many analyses assessing the effect of air power against mobile TBMs. There seems to be a lack of a consensus on whether the TBM defense effort was a success. Regardless, TBM attacks were top stories for the media and a constant concern for coalition forces throughout the war. This high-level exposure will fuel many nations' desire to enhance both their offensive and defensive TBM capabilities.

There are many reasons why nations might aspire to a TBM capability. TBMs provide an ability to attack targets at long range with little notice. Secondly, TBMs are difficult to destroy following launch. Lastly, TBMs are able to carry conventional, chemical, or nuclear ordnance. These capabilities provide a nation with an easy method of inducing instability into world affairs. (Navias, 1990)

The list of countries with known tactical systems is presented as Appendix A (Berhow, 1993). While no nation in particular should be viewed as a potential U.S. adversary, this table certainly illustrates the influence the TBM threat may have on future world affairs. One must also remember that these countries and others are actively seeking to improve their TBM capability.

An example of the current TBM state of affairs was presented by Hair (1993):

> The MAZ-543 transporter-erector-launcher (TEL) is the primary launch and support vehicle for the SS-1B (Scud-A), SS-1C (Scud-B), and SS-12 (Scaleboard) surface-to-surface missiles. The vehicle is 12m long and with the missile it weighs 29,000 kg. It has a crew of 4, is capable of 60kph on hard surfaces and has a range of 1500km. Its 525hp diesel engine and 8-wheel drive system allow it to operate over rugged terrain but at the expense of speed and range.

A "TBM revolution" would promote across the board improvements to TBMs. Missile accuracy will certainly improve. Increasing ranges and payloads will add to the lethality of these weapons. The TELs which serve as the launch platform may become increasingly mobile, diminishing the impact of terrain.

2. Unattended Ground Sensors (UGSs)

The use of an unmanned remote sensing system for the cueing of response forces has a fairly recent history. Development of an unattended ground sensor system was begun during the Vietnam War in 1966 by Secretary of Defense Robert McNamara. The sensors were to be part of a system designed to track movement of enemy forces, termed the "McNamara Line." The sensors' use increased: both for offense and defense. For example, they provided an early warning against surprise attack during Khe Sanh. Attack aircraft were given assignments of targets on the Ho Chi Minh trail based upon sensor reports.

These sensors were largely considered a success by the military, although there are some who would prefer to say "limited" success. (Dickson, 1976)

Unattended ground sensor systems have continued to be developed over the last twenty years. These systems have found a natural home with the Army and Marine Corps. The Army currently uses the Remotely Monitored Battlefield Area Sensor System (REMBASS) and recently the Marine Corps fielded the Tactical Remote Sensor System (TRSS). Both of these systems have the mission of providing information to tactical commanders about the movements of enemy forces within those commanders' areas of interest.

These systems attempt to detect and classify force elements that pass near the sensors. UGSs have the potential to exploit a variety of target signatures, for example: acoustic, infrared, magnetic, optical, and seismic. The classification capability is limited to determining the difference between dismounted infantry and vehicles. A general UGS system consists of its detectors, a transmitter/relay system, and a monitoring station. The detection through relay components are limited by battery lifetime. Research into improved UGS systems continues, but to-date no system has the ability to provide real-time target classification information to on-station attack aircraft. (van Kan, 1994)

The proposals for incorporating UGS systems into attack operations are varied but contain two main themes. The first theme, represented by Berhow (1993), is that the desired function of UGSs is as a warning mechanism in previously determined threat areas. The largest amount of effort is devoted to determining where TEL activity is the highest and placing sensors in those areas. The second theme, contained in the Junker thesis (1995), is

that the focus of the effort should be entirely along roadways and intersections. He suggests that a further research area would be to provide a decision support aid for UGS placement based upon network theory.

A "UGS revolution" would certainly provide enhanced capabilities vital to the counter TBM effort. Extended battery lifetimes would enable sensors to be placed in areas not yet believed to contain TELs. Longer detection ranges and enhanced data transmission ability are capabilities that are certain to develop. The ability of these sensors to correctly classify targets of interest is the most promising capability awaiting to experience its revolution.

III. SIMULATION DEVELOPMENT

A. WARFARE SCENARIO

The purpose of this section is to paint the picture of a warfare scenario which projects the military forces involved into a post-RMA environment. The concept is fashioned after littoral warfare in general, yet applies to many warfare operations. An aggressor nation bordering international waters, seeking to promote regional instability, has begun to actively deploy TELs. U.S. National Command Authority has directed forces to the area in an attempt to deter aggression. Carrier-based attack aircraft arrive in theater with a mission of conducting attack operations. Following a brief prehostility phase, fighting begins.

When modeling theater-level war there are many components and decisions that must be made as to which warfare components are desired. Because of the scope of this thesis, the warfare scenario explained above may be simply viewed as a "cat and mouse" game between attack aircraft and TELs. In order to explore further the implications of Dominant Battlespace Awareness other players will be added to the game. TEL decoys will add a deception element to the scenario for the aggressor nation. A UGS sensor network will increase awareness for the units conducting attack operations.

B. MODEL ASSUMPTIONS

Every model depends heavily on its construction and input parameters and the model to be developed in the following paragraphs is certainly no different. This model does not attempt to predict or quantify *exactly* what outcome to expect. Rather, its results are

designed to provide *insight* into what the author believes are issues that should be included in any discussion of an RMA.

The first set of model assumptions center on the theater of operations themselves. The model assumes that Dominant Battlespace Awareness should cover a 200 mile by 200 mile region (Gordon, 1994). This model further assumes the terrain to be uniform or very nearly uniform (like a coastal plain or desert), thus having minimal impact on the outcome.

There are several key assumptions for the U.S. forces in the scenario. The first is that the concept of DBA allows for attack aircraft to be aware of the presence and state of enemy air defenses. For this reason, attack aircraft do not experience attrition in the present model. Secondly, U.S. dominance at sea is assumed so there is no naval interaction in this model. Finally, the model also does not permit attack forces to follow a strategy of using overhead assets to enhance their knowledge. This includes intelligence gathering during the prehostility phase as well as the identification of TELs immediately following a TBM launch (also known as the "flaming datum strategy").

The most important assumptions pertain to the aggressor nation. This model assumes that this nation has experienced an RMA so that its TELs have improved capabilities. Specifically:

- The aggressor nation has developed a system of hidden logistic bases. This nation is also aware of U.S. desire to exploit existing roadways in their search efforts. Thus, the TELs have enhanced off-road capability. TELs have advanced camouflage abilities and can remain hidden from search efforts while waiting to launch their TBM.

- The TELs operate with an advanced command and control capability. There is no formal command structure for attack operations to exploit. TELs essentially operate independently and their logistic support structure is always able to meet their demands.

- The aggressor nation has complete knowledge of remaining TEL capabilities and will continue to fight until all TELs are destroyed.

- The aggressor nation does not have an ability to reinforce the area with additional TELs.

The last two assumptions can be easily modified in the simulation to provide for TEL reinforcement or a different end of hostilities criteria (such as percentage of TEL force remaining).

C. DESIGN CONSIDERATIONS

Appendix C contains the program used in this thesis. This section highlights the basic functionality of each of the entities represented in the program. The simulation was written in the MODSIM II® , a modular, object-oriented simulation language.

1. Attack Aircraft

Because an assessment of attack aircraft is beyond the scope of this thesis, the attack aircraft do not represent a particular type of aircraft, but rather an attack capability. Aircraft are located on board a stationary aircraft carrier and operate in only two dimensions. There is no explicit modeling of altitudes or attack profiles. Figure 2 is used to illustrate the fundamental properties of aircraft interaction in the simulation. In a basic scenario (one in which UGSs have not been introduced) an attack aircraft will receive an initial search datum determined uniformly at random over the operational area (Op Area). The aircraft will fly

to that point to begin a random search. The aircraft will conduct a search of a circular area and the outcome of the search will trigger one of two responses:

- No Target is found - the aircraft will move forward a distance equal to twice its attack radius and conduct a second search. This motion is modeled to ensure that glimpse regions do not overlap. The search movement is only permitted if the unit has remaining on-station time. When there is no on-station time remaining, the aircraft returns to the carrier. The pattern of an aircraft's search is referred to as a *forward 2-R random search.*

- A Target is Found - the aircraft conducts one attack and returns to the carrier. The determination of a successful attack is encapsulated as a single probability of success which is constant throughout the simulation.

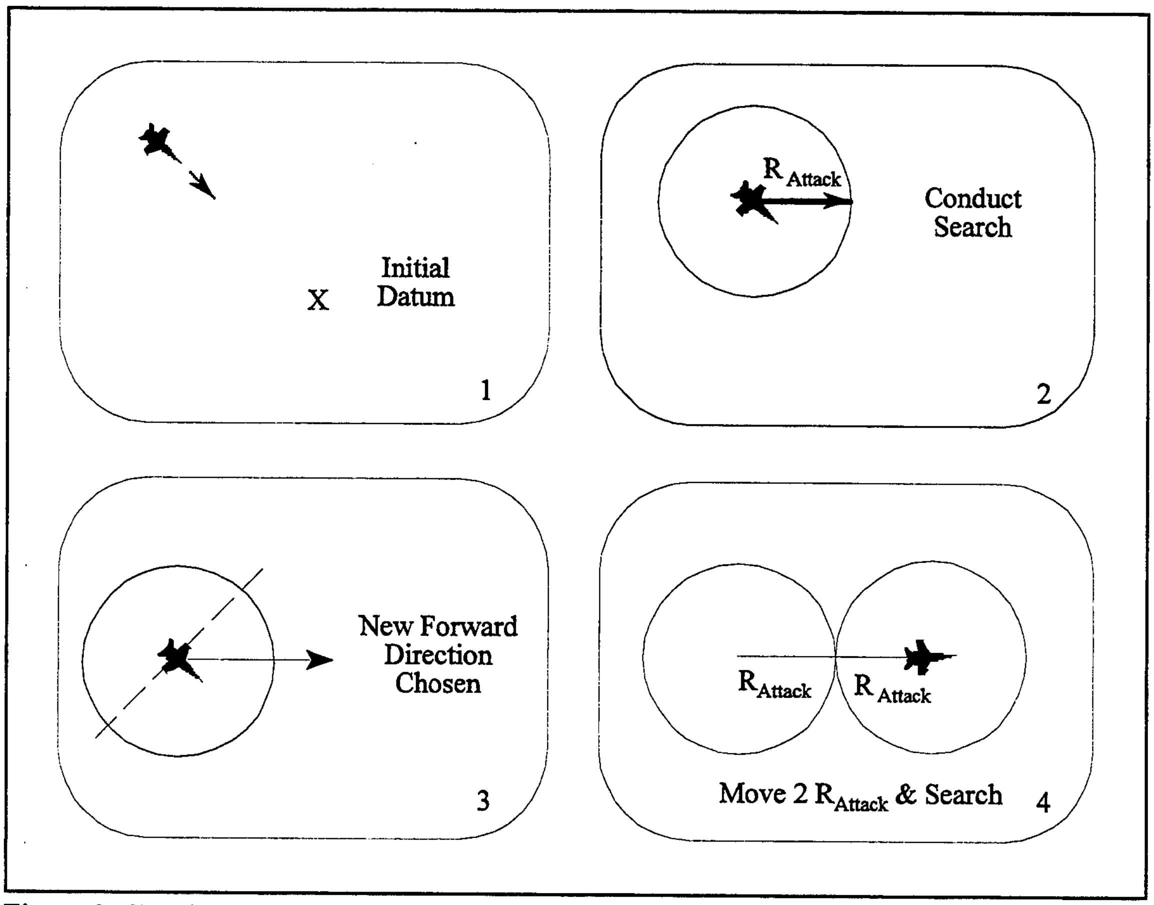

Figure 2. Fundamental Procedure for Aircraft Interaction.

Upon return the aircraft must wait for a turnaround time to expire before obtaining its next assignment. While this certainly is not the only possible rules for aircraft actions, but the aircraft in this model have a primary purpose of force attenuation. Research into improved search tactics with UGS networks and bomb damage assessment are beyond the scope of this thesis.

In a scenario in which UGS networks are present, the aircraft can receive target assignment either on takeoff from the carrier or during flight. An aircraft which returns to base with pending assignments transfers those assignments (without a transmission time delay) to the aircraft which is closest to the datum.

It is important to emphasize that the aircraft in this simulation are designed to operate independently from other aircraft. This feature was produced intentionally in order to approximate a continuous coverage capability, an assumed feature of a 'Dominant' system. When aircraft are first created, each receives the same values for fundamental characteristics such as on-station time. This deterministic approach coupled with the methodology of returning to the carrier after a single attack, induces cyclic behavior in aircraft attack frequency. This issue will be further explained in Chapter IV, when the simulation is compared to two analytical models.

2. UGS Network

The most important characterization of the unattended ground sensors is that *sensor networks are modeled, not individual sensors.* Thus a detection radius of one mile, for example, means that groups of sensors have been placed along a road network and the total

area covered by each group of these sensors is a circle with a radius of 1 mile. However, it is important to note that the model does not explicitly model a transportation network. Figure 3 is provided to illustrate the UGS concept.

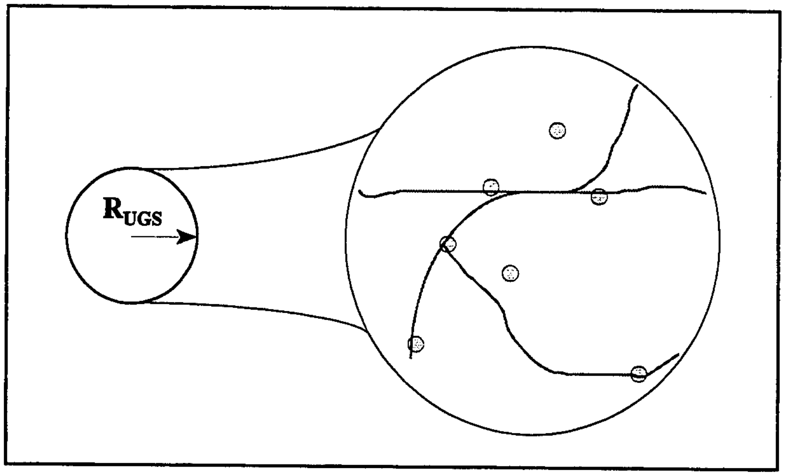

Figure 3. UGS Representation.

The UGSs are given the ability to perfectly discriminate infantry or tanks from TELs with certainty; thus false detections are not modeled. The aggressor nation's TEL decoys cannot be distinguished from real TELs by the UGS network. Upon detection, the UGS will transmit its location to the aircraft squadron commander, which will pass that detection, without a time delay, to the closest on-station aircraft. If no aircraft are in flight over the area, the detection is tagged with a time and placed into a target queue on the carrier. As aircraft takeoff they receive targets from the queue which are not time-late. Time-late targets are discarded and the aircraft receive a fresh assignment. The aircraft then proceeds as before to begin its forward 2-R random search.

3. Transporter Erector Launcher Activity Cycle

The TELs in the model follow a simple cycle of action presented in Figure 4. TELs are first given an initial starting location selected randomly from the model-generated list of possible logistic bases. Then a hiding location is chosen within the Op Area with the perspective coordinates randomly and independently drawn from a uniform distribution within the area boundaries. The TEL travels to that location at its normal speed. While traveling to this location, the TEL is visible to attack aircraft. The TEL is then perfectly camouflaged upon arrival at the hiding location. After waiting for a random time determined by a waiting distribution, the TEL emerges and launches a TBM. The TEL then returns to a logistic base at evasion speed. The TEL is hidden from attack aircraft while reloading, and the process begins again.

4. TEL Decoys

TEL decoys move randomly throughout the search area at a TEL's normal speed. They do not experience periods of invisibility. Their job is to trigger UGS response, and provoke attack by aircraft. They are essentially "asset sponges."

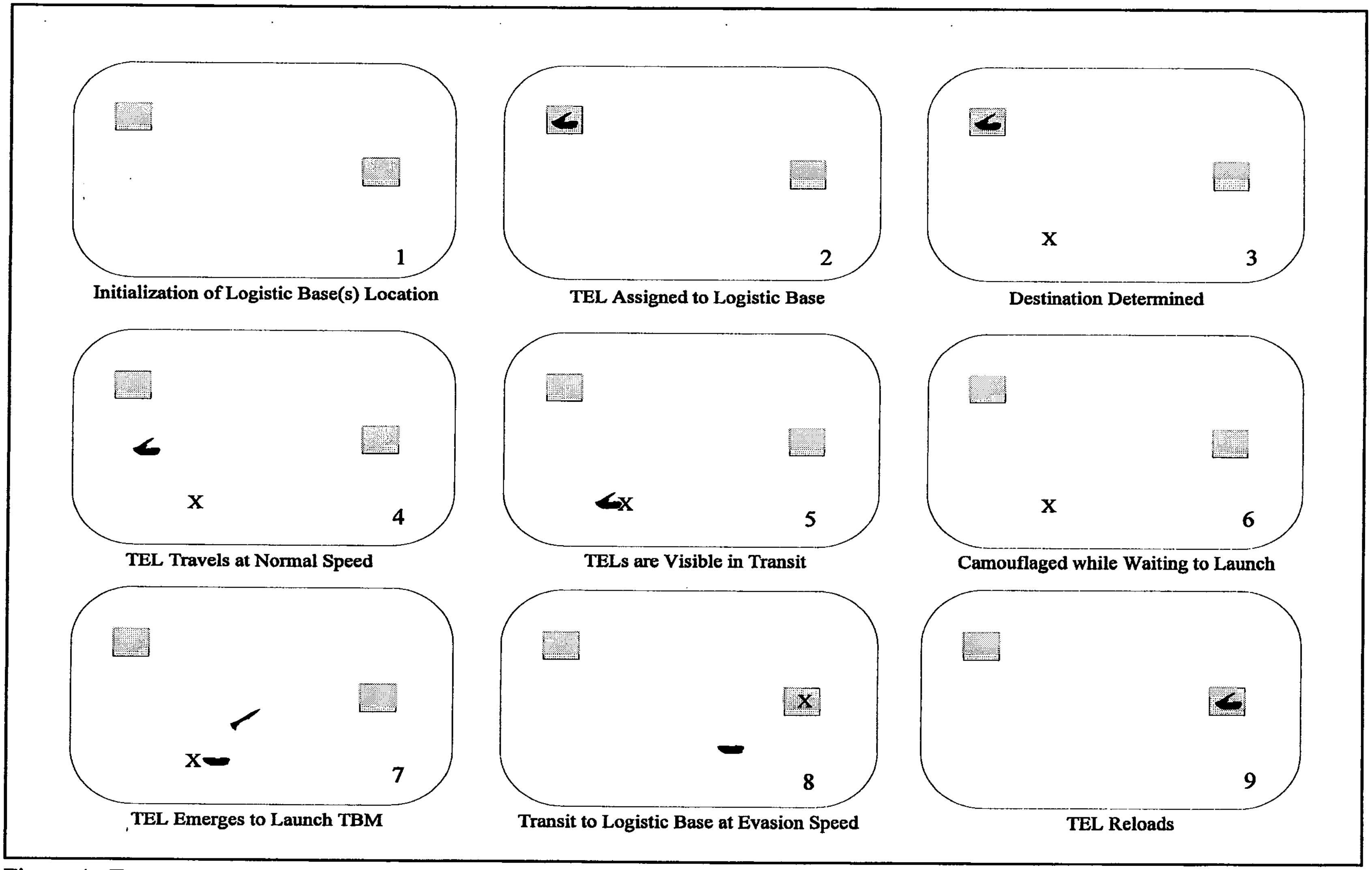

Figure 4. Transporter-Erector-Launcher (TEL) Activity Cycle.

D. INPUT PARAMETERS

There are several descriptive parameters used by the model. These parameters are not intended to be totally faithful to reality, only to provide a general sense of it. Yet, they do provide a reasonably detailed description of the scenarios considered in this thesis. Table 1 illustrates the primary components and their associated parameters:

Component	Associated Parameters
Attack Aircraft	Number of Attack Aircraft Location of CV Probability of Kill Attack Velocity Attack Radius On-Station Time Turn-Around Time Estimated TEL Speed
Unattended Ground Sensor Network	Sensor Range Number of Sensor Networks
Transporter Erector Launchers (TELs)	Number of TELs Normal Speed Evasion Speed Hiding Time Parameters Reloading Time Parameters Minimum Number of Logistic Bases Logistic Base Factor
TEL Decoys	Number of Decoys Normal Speed
Operating Area	Range of Boundaries Number of Days in Prehostility Phase

Table 1. Simulation Model Input Parameters.

IV. DISCUSSION OF RESULTS

Before beginning a detailed discussion of the results of the model an attempt should be made to gain insight into the computer simulation. Two simplified and computationally tractable analytical models based upon the methodology of the simulation are formulated. The analytical models and the simulation are contrasted and compared. The simulation is then used to generate scenario outcomes.

The simulation is discussed in terms of four comparison cases. First the model is run with aircraft searching for TELs. The second case adds decoy TELs that remain visible throughout the simulation in an attempt to absorb attacks from the aircraft. In the third case the decoy TELs are not present and the aircraft search for TELs with the assistance of a UGS network. Finally, all four components are included (TELs, decoy TELs, aircraft and UGSs) in the final comparison case. The comparison cases are summarized in Table 2.

	Case 1	Case 2	Case 3	Case 4
Aircraft	✔	✔	✔	✔
TELs	✔	✔	✔	✔
Decoy TELs		✔		✔
UGSs			✔	✔

Table 2. Comparison Cases for Simulation.

Examination of the output from the analytical model or the simulation is in terms of two measures of effectiveness (MOEs). The first is based on the length of time until the aggressor nation chooses to cease hostilities. In this model the TELs continue to conduct

attacks until all TELs have been destroyed. The time from the end of the prehostility phase (during which the TELs exhibit aggressive behavior but do not launch TBMs) until the last TEL is destroyed will be called the *survival time*. Thus, the first MOE provided will be an estimator of the expected survival time. The second MOE is based on the total number of TBMs launched by the TELs during hostilities. The simulation will determine an estimator of the expected total number of TEL launches until the last is destroyed.

A. ANALYTICAL MODELS

The simulation designed for this thesis was done with the specific purpose of examining issues within the concept of dominant battlespace awareness. Laying the groundwork for the results that follow is best done when the results are first compared against a realization of what one might expect to happen. The analytical models presented here will provide a measure of validity and serve as bounds on the functional form of the MOE's.

1. The Independent Markovian Model

Assume B aircraft search a region of size A square miles seeking to destroy R TELs. The aircraft operate independently from one another and follow a random search pattern. At each instant of time a circular area is searched with a given attack radius and upon detection the target is destroyed with some probability, *p*. In continuous time assume that the probability an aircraft discovers and subsequently destroys a TEL in time $(t, t + dt)$ is equal to $\lambda(B)\, R\, dt$. Let $\lambda(B)$ be the rate of TEL destruction, a linear function of the number of aircraft searching (the specifics of determining λ are presented in the pages to follow). Also

assume that only one TEL destruction may occur in any (short) period *of length dt* and that the probability of TEL destruction in these periods are independent. We would now expect that the time for the first TEL to be destroyed will occur according to an exponential distribution with mean $\frac{1}{\lambda(B)\cdot R}$. Now there are (R-1) TELs remaining. Then under the current assumptions the next TEL will be destroyed after an independent exponential time with mean $\frac{1}{\lambda(B)\cdot(R-1)}$. This process will continue until the last TEL is destroyed after exponential time $\frac{1}{\lambda(B)}$. Let random the variable T_i represent the time for the destruction of the *i*th TEL. The variable T represents the total time required for the destruction of all TELs. Thus, the random variable T can be expressed as

$$T = T_1 + T_2 + T_3 + \ldots + T_R \; . \tag{1}$$

Since T is a sum of exponential random variables its expected value is simply determined:

$$E[T|B,R] = E[T_1] + E[T_2] + E[T_3] + \ldots + E[T_R] \tag{2}$$

$$E[T|B,R] = \frac{1}{\lambda(B)} \left[\frac{1}{R} + \frac{1}{R-1} + \ldots + 1\right] . \tag{3}$$

The right-hand side of this equation approaches $\frac{1}{\lambda(B)}\ln(R)$ as R becomes large. The functional form of this equation is such that the expected time is a monotonically increasing function but with decreasing marginal rates as R increases. This expected shape of the survival time curve will resurface when this is compared to simulation output.

The analytical model described also has basic functionality when seeking to describe the expected total number of TBM launches conducted by TELs during hostilities. Adhering to the same set of assumptions for the destruction rate, assume that TELs launch TBMs according to a launch rate $1/\tau$. Then with L_i representing the number of launches from the ith TEL, $E[L_i] = \frac{1}{\tau}\cdot\frac{1}{\lambda(B)}$. Letting L reflect the total number of TBM launches,

$$E[L] = E[L_1] + [L_2] + \ldots + [L_R] = \frac{R}{\tau\cdot\lambda(B)} \,. \tag{4}$$

The expected total number of launches as shown above is linear with respect to R.

Of primary concern at this point is the determination of the destruction rate, $\lambda(B)$. It is likely that $\lambda(B) = \lambda_{KILL} \cdot B$ to a very good approximation, where λ_{KILL} depends on model parameters. Based upon the actions of the unit components from the simulation there are 3 parameters that will be used to construct the kill rate. These parameters are

- γ_{Detect} - Detection Rate.
- $P_{KILL|Detect}$ - Probability of Kill given TEL has been detected.
- r_v - Long Run Probability TEL is visible.

Determining γ_{Detect} depends upon the methodology of aircraft movement within the simulation. Each aircraft has a flight cycle consisting of three parts: transit between the carrier and the operational area (Op Area), searching for TELs on station in the Op Area, and a transitional period following the aircraft's return to the carrier. Designating the total time

to complete this cycle as C, aircraft cycle time is expressed as

$$C = T_{OnStation} + T_{BackandForthTravel} + T_{CVTurnAround} \; . \qquad \textbf{(5)}$$

During this cycle time, while on-station over the region, the aircraft performs its *forward 2-R random search.* The search algorithm permits a single discrete "cookie cutter" glimpse before initiating the on station time. Thus during its on station time an aircraft (searching with velocity, $v_{a/c}$) searches N circles with an attack radius r_{Attack}. The area visited during a cycle is found

$$N = \left[1 + \frac{T_{OnStation}}{\left(\frac{2 \cdot r_{Attack}}{v_{a/c}}\right)} \right] \qquad \textbf{(6)}$$

$$A_{Cycle} = N \cdot \pi r^2_{Attack} \qquad \textbf{(7)}$$

$$A_{Cycle} = \left[1 + \frac{T_{OnStation}}{\left(\frac{2 \cdot r_{Attack}}{v_{a/c}}\right)} \right] \cdot \pi r^2_{Attack} \; . \qquad \textbf{(8)}$$

The A_{Cycle} is then divided by the total area, A_{Total} to obtain the coverage factor, CF. In order to transform the dimensionless coverage factor into a coverage rate divide by the cycle time, C. Thus $\gamma_{Detect} = \frac{CF}{C}$.

The probability of kill given detect, $P_{KILL \mid Detect}$, is an input parameter. If a TEL is within the attack radius of the aircraft and is visible the aircraft detects the TEL. The proportion of time the TEL is visible is a function of the time the TEL is transiting between logistic and hiding sites. If the true parameters were known

$$r_v = \frac{T_{Moving}}{T_{Moving} + T_{Hidden}} . \tag{9}$$

The time the TEL is moving depends on the simulation methodology and input parameters. The TEL is hidden during the waiting period to launch a TBM and the reloading period at a logistic base. The three parameters are combined to produce the kill rate as

$$\lambda_{KILL} = \gamma_{Detect} \cdot p_{KILL \mid Detect} \cdot r_v . \tag{10}$$

As mentioned previously the analytical model also depends on the determination of the launch rate, τ. An individual TEL launch cycle has four components:

- Visit to Logistic Base (Initial or Reload)
- Travel to Hiding Point
- Waiting to Launch TBM
- Travel to Logistic Base

These components occur in a time that is directly determined randomly from a distribution (logistic base visits and waiting periods) or a time that depends upon the randomness of the

simulation locations (travel times). An estimate of this launch cycle time is found to be

$$E[\tau] = E[T_{Reload}] + E[T_{TraveltoHide}] + E[T_{Waiting}] + E[T_{TraveltoBase}] \quad (11)$$

2. The Wave Model

This model assumes that the aircraft conduct attack operations as a group or wave of B aircraft. The aircraft search a region of size A square miles seeking to destroy R TELs. An individual aircraft with an attack radius, r_{Attack}, attack velocity, $v_{a/c}$, and an on-station time, $T_{On\ Station}$, searches the same number of circles, N, was first presented in Equation 6. Thus, the B aircraft in the wave search n sub-regions found as

$$N = \left[1 + \frac{T_{OnStation}}{\left(\frac{2 \cdot r_{Attack}}{v_{a/c}}\right)} \right] \quad (12)$$

$$n = N \cdot B \, . \quad (13)$$

The B aircraft have a probability of killing a TEL found in one of the n regions that depends upon several other probabilities. First, the TEL must be present in the sub-region begin searched. Letting t represent the number of sub-regions available in the region,

$$t \approx \frac{A}{\pi r^2_{Attack}} \, . \quad (14)$$

The probability any one of R TELs is present in an aircraft's search sub-region is $P_{Present}(R) = \frac{R}{t}$ (this assumes R is small enough compared to t, and that no two (or more) TELs are present in a sub-region. The probability a TEL is visible given it is present is determined as

$$r_{Visible\,|\,Present} = \frac{T_{Moving}}{T_{Moving} + T_{Hidden}} \; . \qquad \textbf{(15)}$$

The TEL must also be detected once visible, represented by $P_{Detect\,|\,Visible}$. Finally an aircraft destroys a detected TEL with probability $P_{Kill\,|\,Detect}$. These supporting probabilities are used to produce the probability a TEL is killed,

$$P_{Kill}(R) = P_{Kill\,|\,Detect} \cdot P_{Detect\,|\,Visible} \cdot r_{Visible\,|\,Present} \cdot P_{Present}(R) \; . \qquad \textbf{(16)}$$

The probability that none of R TELs are killed during a wave's on-station time is

$$\beta(R) = [1 - P_{Kill}(R)]^{n} \; . \qquad \textbf{(17)}$$

Now let T_K be the time until some TEL is killed and let m(R) represent $E[T_K\,|\,R]$; note that this is conditional on there begin R TELs present initially. One would expect this time to reflect the probability of killing a TEL as well as the time the wave spends in transition to the Op Area, T_A. Assuming that when a TEL is killed the entire wave returns and that $T_{On\ Station} / 2$ is the mean time the TEL is killed given the TEL will be killed,

$$m(R) = T_A + (T_{OnStation} + m(R))\cdot\beta(R) + \frac{T_{OnStation}}{2}\cdot(1 - \beta(R)) \ . \qquad \textbf{(18)}$$

Solving for m(R),

$$m(R) = T_A + \frac{T_{OnStation}\cdot\beta(R)}{1-\beta(R)} + \frac{T_{OnStation}}{2} \ . \qquad \textbf{(19)}$$

With the foundation built by the previous equations, T, the expected time to kill all R TELS is found as

$$T = m(R) + m(R-1) + m(R-2) + \ldots + m(1) \ . \qquad \textbf{(20)}$$

An analytical estimate for the second MOE, the expected number of launches may be formulated in several steps. Retaining m(R) as the expected time for any of the R TELs to be destroyed by the wave, the expected number of TBM launches from any individual TEL, during m(R) depends upon the launch cycle time, τ, and equals $\frac{m(R)}{\tau}$. Since there are R TELs, the expected number of launches is $\frac{R\cdot m(R)}{\tau}$. Each successive TEL destruction reduces R while τ is assumed to remain constant. Finding the expected total number of TBM launches, L, is found to be

$$E[L] = \frac{R\cdot m(R)}{\tau} + \frac{(R-1)\cdot m(R-1)}{\tau} + \frac{(R-2)\cdot m(R-2)}{\tau} + \ldots + \frac{m(1)}{\tau} \ . \qquad \textbf{(21)}$$

The two models formulated in the previous sections each represent a different methodology for the aircraft conducting attack operations against TELs. The first model reflected a continuous random search by aircraft operating completely independently of one another. This model will be referred to as the "independent model." The second model, termed the "wave model," represented aircraft operating as a wave. Each of these models will serve to bound the simulation results as is shown in the next section. The two models are based on assumptions that, in effect, surround the way in which the simulation is now programmed to function.

B. COMPARISON OF SIMULATION AND ANALYTICAL MODELS

The analytical models will be used to generate results based upon the set of parameters shown in Table 3.

Parameter	Value
Number of TELs	1 to 19 by 3
TEL Normal Speed	15 mph
TEL Evasion Speed	25 mph
Mean Hiding Time	8.0 hours
Mean Reload Time	3.0 hours
Min Number of Logistic Bases	1
Logistic Base Factor	5
Number of Aircraft	1 to 9 by 2
$Prob_{KILL \mid Detect}$	0.9
Aircraft Velocity	400 mph
Attack Radius	10.0 miles
On Station Time	2.0 hours
Turn Around Time	3.0 hours
Op Area Size	200 x 200 sq. miles
CV Location	125 miles off coast

Table 3. Input Parameters for Analytical Model.

The input parameters of Table 3 are used in accordance with the previous discussion to determine the desired analytical model values, with two exceptions. The expected TEL travel times and the aircraft back and forth times are dependent upon distances between random locations. The distributions of these distances are not easily determined. Estimators

of their expected value can be easily obtained by running a separate mathematical simulation. Figure 5 presents, as an example, the results of one such simulation. The simulation was conducted with the mathematical analysis program, MATLAB®.

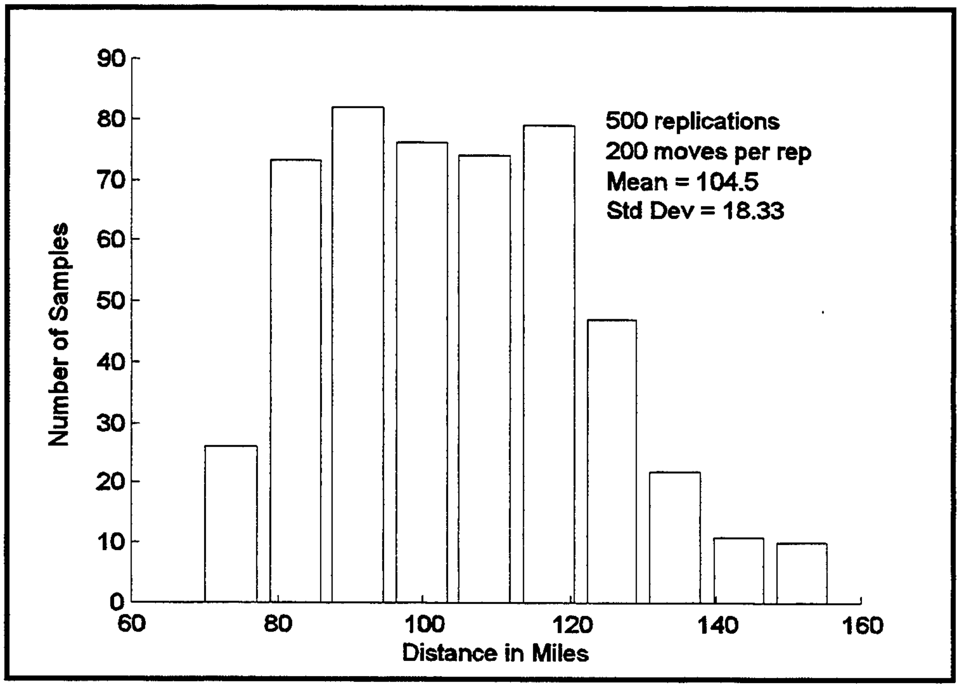

Figure 5. Simulated Distribution of Mean Travel Distance.

Continuing the example, an estimate for $E[T_{Travel\ toHide}]$ from Equation 11 is

$$E[T_{Travelt\ oHide}] \approx \frac{E[Mean\ Travel Distance]}{TEL\ Normal Speed} \tag{22}$$

Using the mean TEL travel distance of 104.5 shown in Figure 5, the expected value of launch cycle time, τ, is found as

$$E[\tau] \approx 3.0 + \frac{104.5}{15.0} + 8.0 + \frac{104.5}{25.0} = 22.15 \ . \tag{23}$$

Table 4 illustrates the values calculated in accordance with the previous equations from the input parameters for the independent model. Those parameters that are dependent upon the number of TELs or the number of aircraft are not included in this table.

Parameter	Calculated Value
$E[T_{\text{Back and Forth Travel}}]$	5.625 hours
C	6.125 hours
N	41 circular search areas
A_{Cycle}	128880.53 sq. miles
CF	0.322
γ_{Detect}	0.0526
$P_{\text{KILL} \mid \text{Detect}}$	0.9
r_V	0.503
λ_{KILL}	0.0238
τ	22.15

Table 4. Independent Model Calculated Parameters.

Table 5 displays the calculated values for the wave model. The probability of detection given a TEL is visible is assumed to be 1.0. The parameters that are dependent upon the number of TELs or the number of aircraft in the wave are not included in Table 5.

Parameter	Calculated Value
N	41 circular search areas
t	127.32 sub-regions for TELs
$r_{\text{Visible} \mid \text{Present}}$	0.503
T_A	4.0 hrs
τ	22.15 hrs

Table 5. Wave Model Calculated Parameters.

The analytical model and the simulation results are compared using the previously defined MOE's: expected survival time and expected total number of launches. Figures 6, 7, and 8 present results for the simulation and the analytical models across several levels for number of aircraft and TELs. The functional form of the analytical models and the simulation are similar and the results agree for low numbers of TELs or aircraft. As TELS increase the expected survival time of the last TEL is greatly affected when one aircraft is present. There is a lesser effect from TEL increases as the number of aircraft is increased.

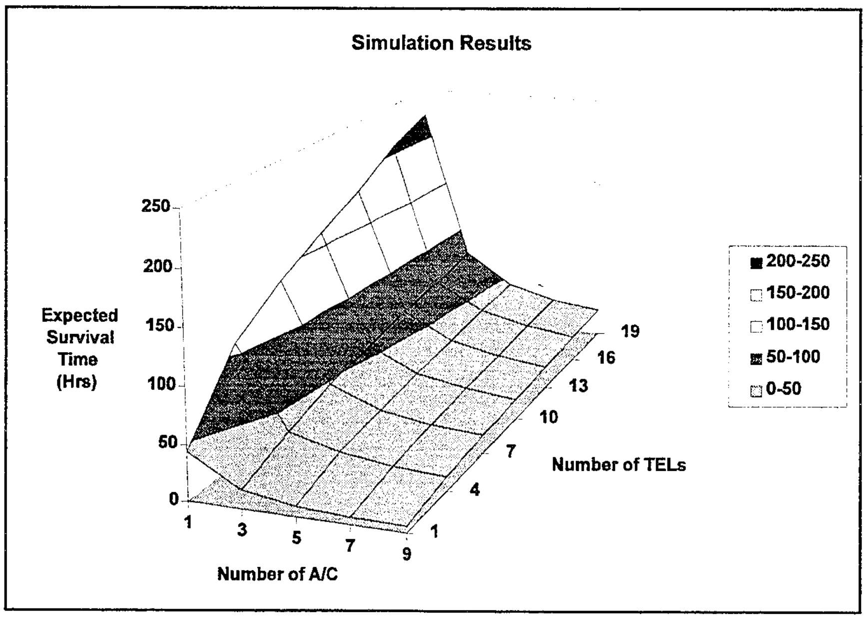

Figure 6. Three-Dimensional Surface Plot of the Expected Survival Time using Simulation Results.

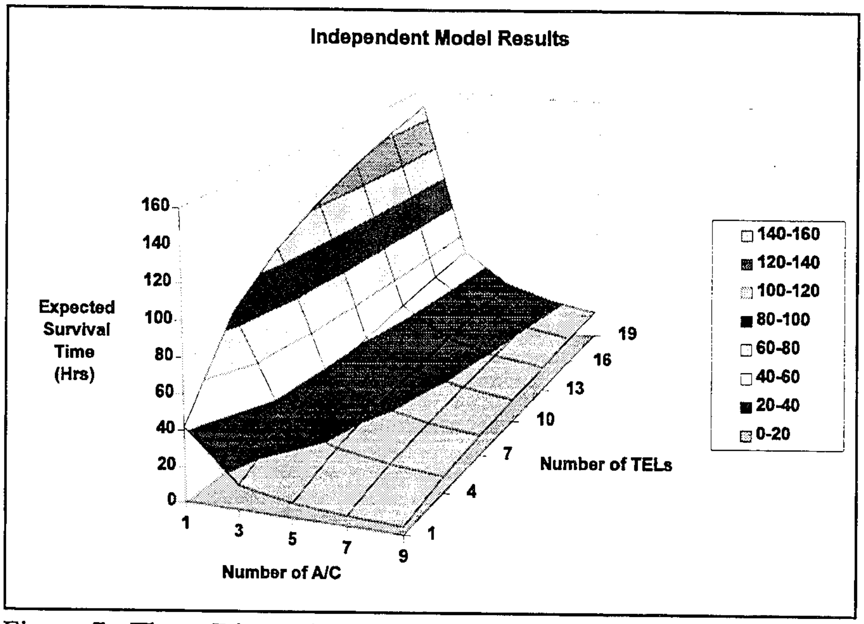

Figure 7. Three-Dimensional Surface Plot of the Expected Survival Time based upon the Independent Model.

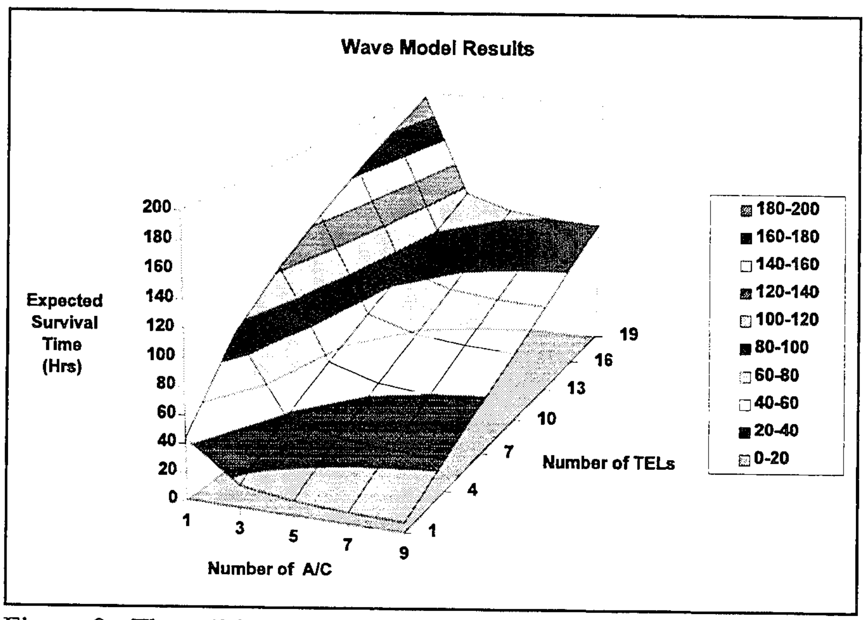

Figure 8. Three-Dimensional Surface Plot of the Expected Survival Time based upon the Wave Model.

Figure 9 provides a two dimensional view for three aircraft against increasing levels of TELs. Recall that the purpose of this comparison is to provide a sense of the functional form of the simulation results. The simulation does exhibit a decreasing marginal return for additional hostile forces if those forces measure their success by the length of the TBM campaign. The analytical models provide an upper and lower bound for the simulation.

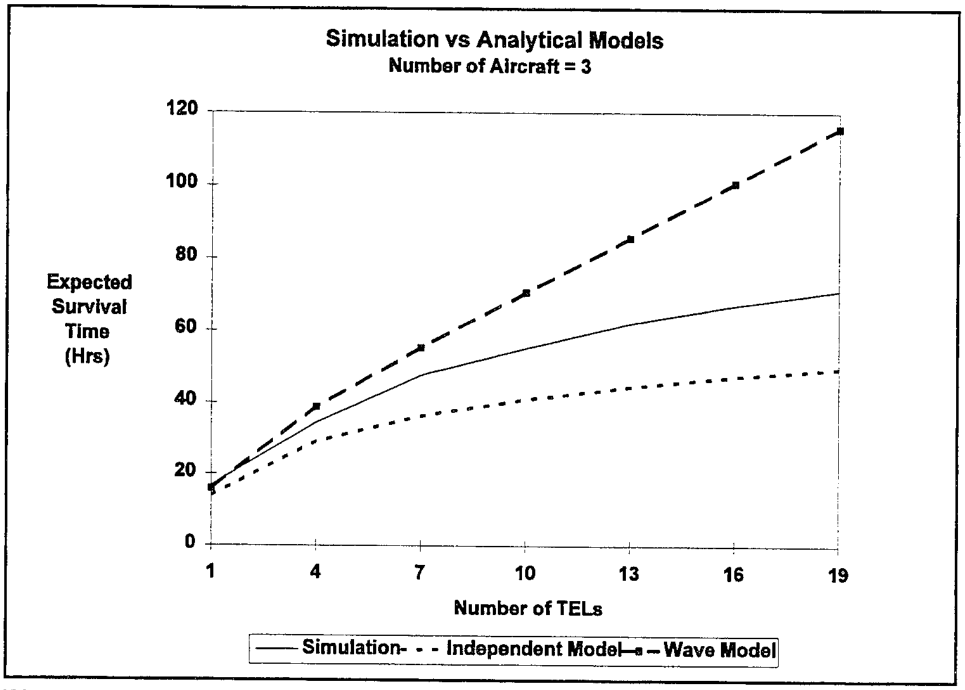

Figure 9. Comparison of Simulation versus Analytical Models for Fixed Level of Three Aircraft.

Although TEL forces experience a decreasing marginal return for survival time with the addition of TELs into the region, Figures 10, 11, and 12 illustrate the impact of more TELs into the region on the total number of launches. The analytical models and the simulation results have similar functional forms.

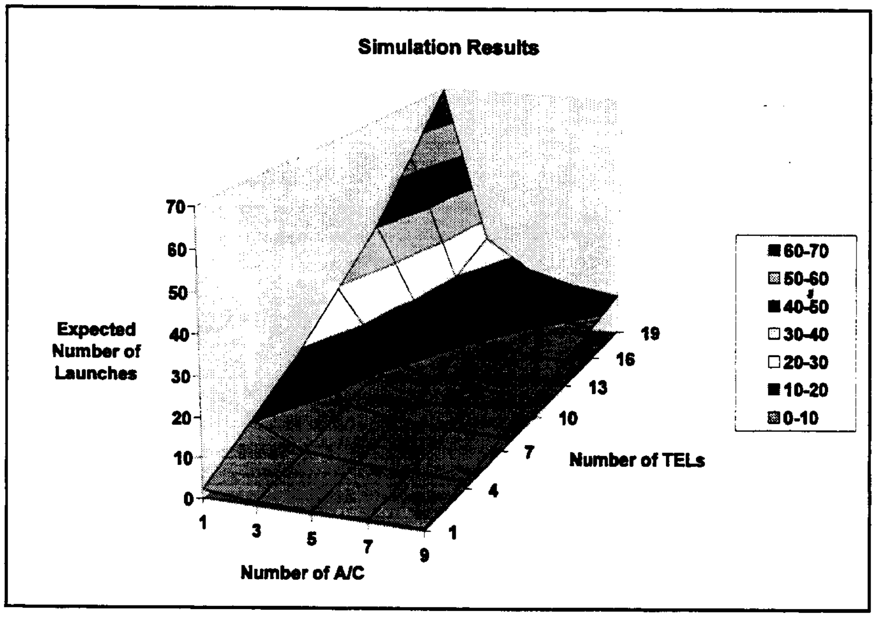

Figure 10. Three-Dimensional Surface Plot of the Expected Number of Launches from the Simulation.

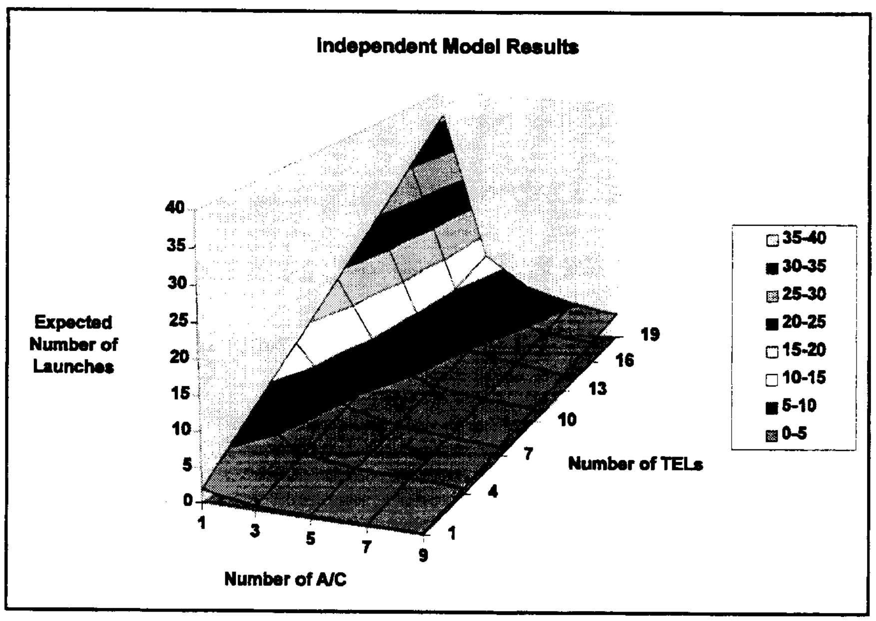

Figure 11. Three-Dimensional Surface Plot of the Expected Number of Launches based upon the Independent Model.

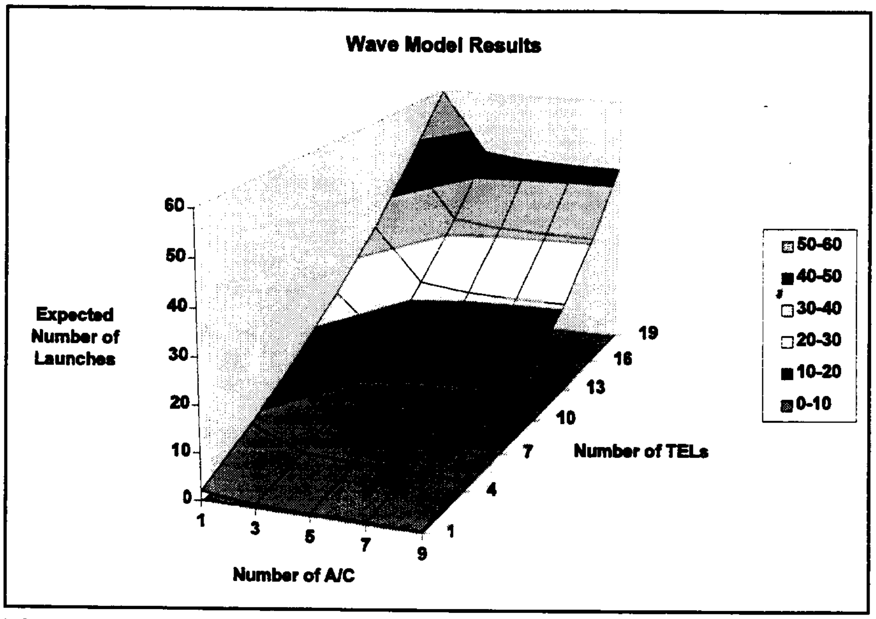

Figure 12. Three-Dimensional Surface Plot of the Expected Number of Launches based upon the Wave Model.

Figure 13 compares the expected number of TBM launches from the simulation against the analytical models. Again there is close agreement for small numbers of TELs and aircraft. The plot of the simulation results shows an increasing marginal return as TELs are increased. The wave model has a curve with a similar form. This phenomenon seems to indicate that the aircraft system experiences diminishing effectiveness as the number of potential targets is increased. The analytical models again serve to bound the simulation results above and below and provide a measure of construct validity. The functional form for the number of launches is an important part of the trend analysis in the pages that follow.

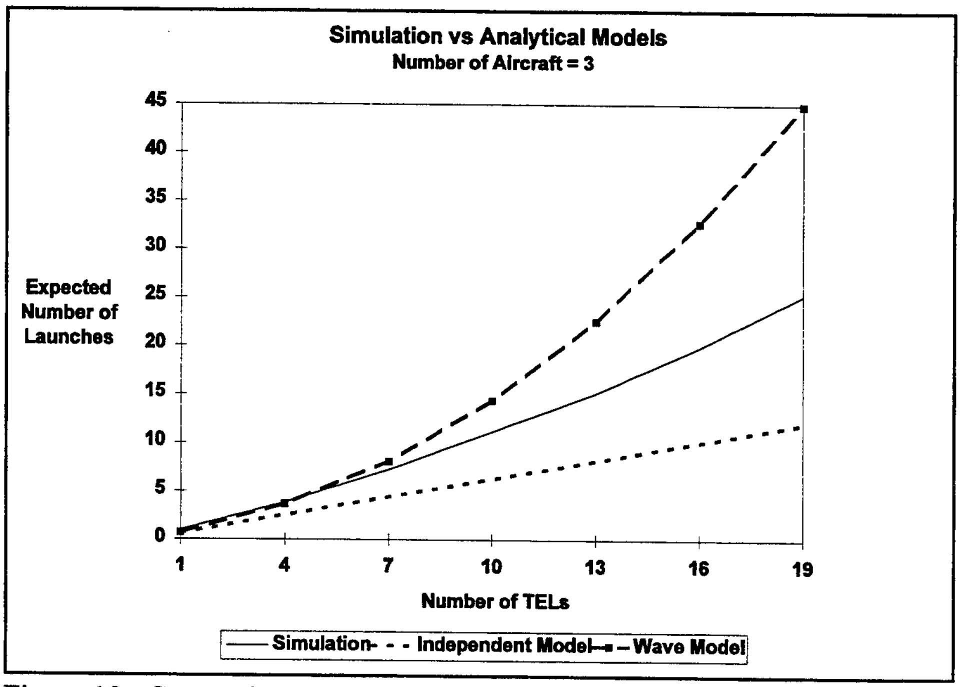

Figure 13. Comparison of Simulation versus Analytical Models for Fixed Level of Three Aircraft.

C. CASE COMPARISON

As mentioned earlier there are four cases used in this thesis. Each simulation case uses the same set of input parameters as provided in Table 6. When decoy TELs are included (cases 2 and 4), the level of decoys is fixed at 10. UGS networks are fixed at 200 in cases 3 and 4. Three-dimensional surface plots for all comparison cases are provided as Appendix B.

Component	Associated Parameters	Value
Attack Aircraft	Number of Attack Aircraft Location of CV (x, y) Probability of Kill Attack Velocity Attack Radius On Station Time Turn Around Time Estimated TEL Speed	1 to 9 by 2 (100.0, 325.0) 0.9 400 mph 10 m 2 hrs 3 hrs 25 mph
UGS Network	Sensor Range Number of Sensor Networks	1.25 m 200 (Case 3 & 4)
TELs	Number of TELs Normal Speed Evasion Speed Hiding Time Parameters Reloading Time Parameters Minimum Number of Logistic Bases Logistic Base Factor	1 to 19 by 3 15 mph 25 mph (4.0, 8.0, 12.0) hrs (1.0, 2.0, 3.0) hrs 1 5
TEL Decoys	Number of Decoys Normal Speed	10 (Case 2 & 4) 15 mph
Operating Area	Range of Boundaries Number of Days in Prehostility Phase	200 x 200 miles 7 days

Table 6. Input Parameters for Comparison Cases 1 - 4.

1. Case 1 versus Case 2

This comparison is between two Cases where the difference lies in the presence of decoy TELs in the simulation. Case 1 is the base Case throughout all future comparisons as it has neither decoy TELs nor a UGS network. Figure 14 shows the expected survival time for Case 1 and Case 2 for aircraft fixed at three. One would expect that the presence of deception would cause an increase in the survival time as Figure 14 illustrates. It is interesting to note that the effect is fairly constant across the number of TELs. Also of note is the decreasing marginal return in the expected survival time for additional TEL assets.

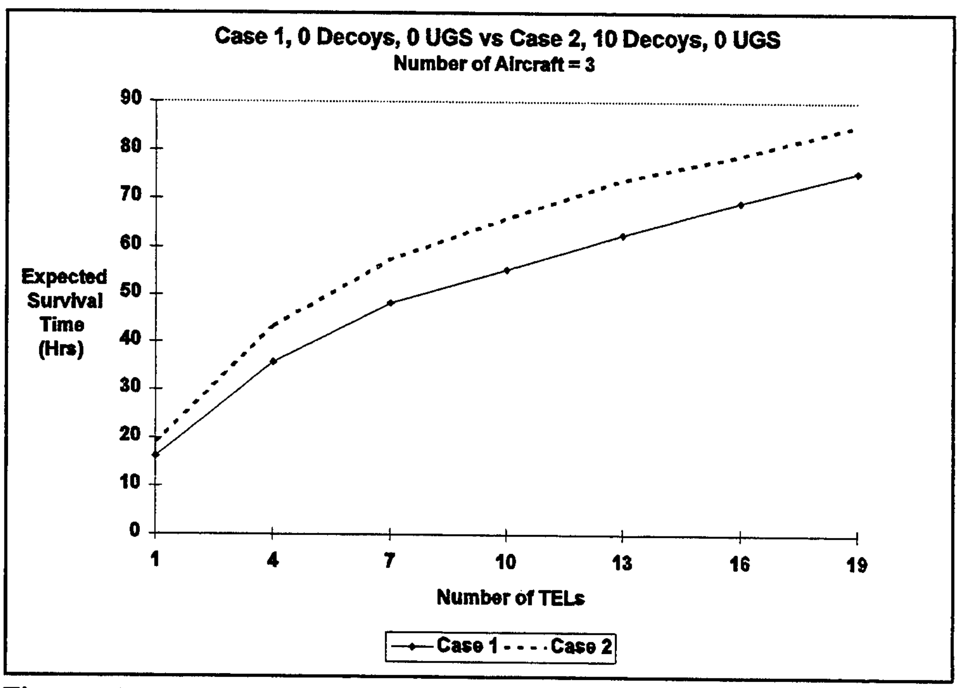

Figure 14. Expected Survival Time. Case 1 versus Case 2.

Figure 15 presents the comparison of the expected number of launches. As in the previous section, there is a payoff for additional TELs in theater. Another observation is that the presence of decoys seems to have a greater effect with more TELs present. Recall that these decoys are indistinguishable from real TELs. Thus they fulfill their designed purpose of "asset sponges." This is a theme that will surface in all the comparisons. A revolution in military affairs that increases an opponent's ability to deceive reduces the advantage U.S. forces seek by improving our own technology.

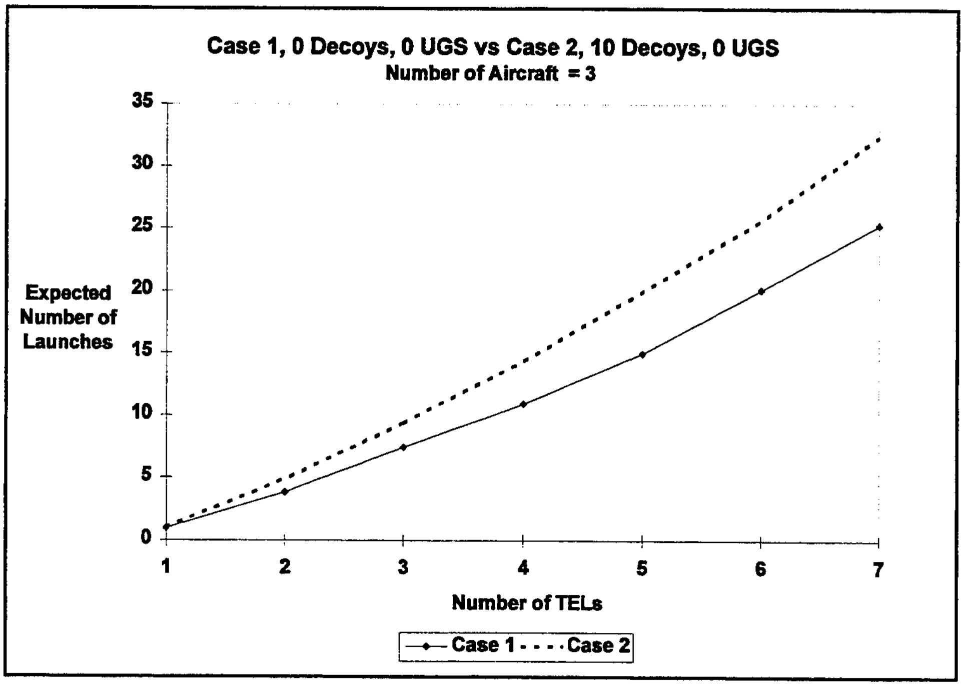

Figure 15. Expected Number of Launches, Case 1 versus Case 2.

What options are open to U.S. forces to counter the TELs of this simulation? The concept of increasing performance while decreasing the number of assets is in contrast with the information provided by Figures 16 and 17. These charts are a cross-sectional view of

the results presented in Appendix B with the number of TELs fixed at seven. There are two areas of interest that these graphs bring to light. The first is that as the number of aircraft in the scenario is decreased there is an increasing marginal return to the TELs for both survival time and number of launches. It is also worth noting that both survival time and number of launches are greater for case 2 (when decoys TELs are present); this is not a surprising result. The effect of this increase is diminished as more aircraft are introduced into the simulation. The tradeoffs between increased capability and increased numbers of forces is an important area for further research, and are not resolvable within the confines of this thesis effort. The impact of deception upon the aircraft system performance begins to shed light on the potential weakness of a nation blindly following the road of technology.

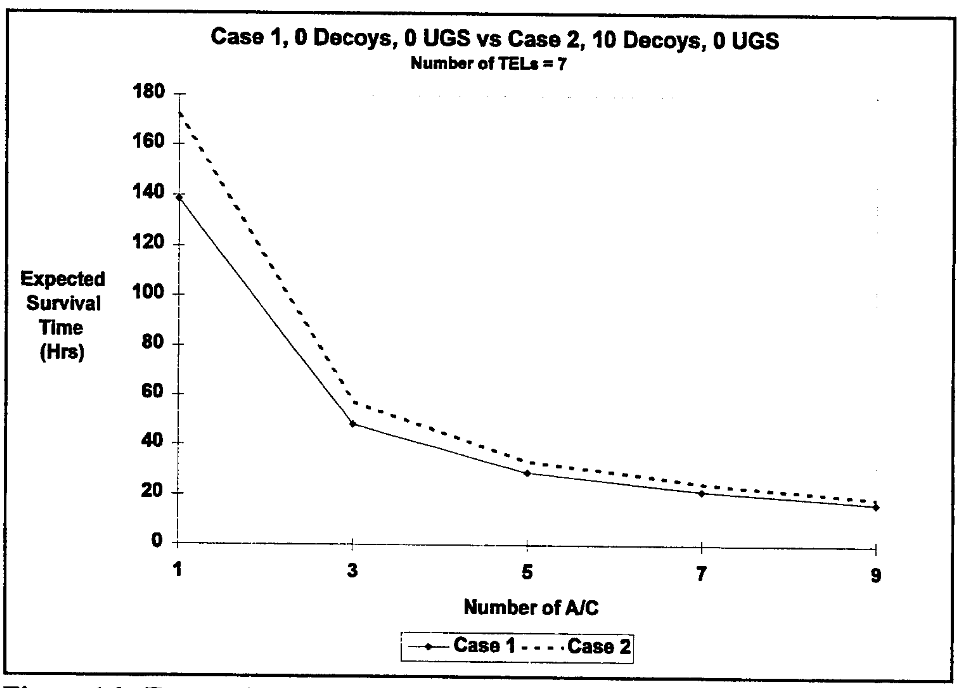

Figure 16. Expect Survival Time, Case 1 versus Case 2. The perspective is for a fixed number of TELs at seven and increasing aircraft.

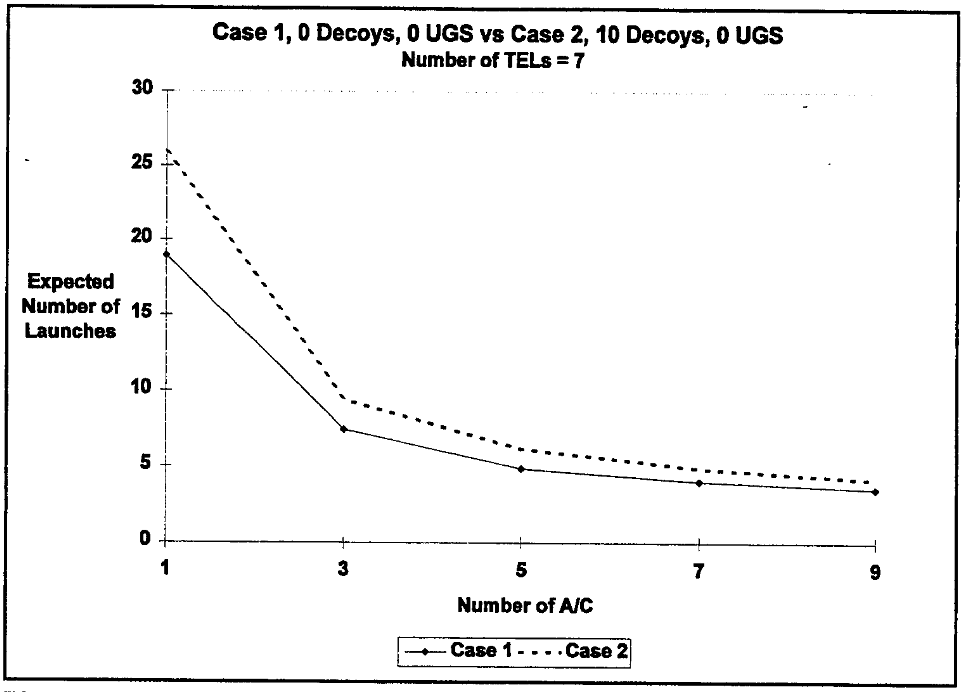

Figure 17. Expected Number of Launches, Case 1 versus Case 2.

2. Case 3 versus Case 4

Case 3 and Case 4 are used to provide insight into the ability of deception to benefit TEL forces when aircraft receive real-time cueing from a UGS network. Figure 18 displays a graph of the expected survival time. Again deception has the ability to extend the conflict even with the capabilities in accordance with the DBA concept. Figure 19 provides another view of the expected number of launches from the simulation. The TELs of Case 4 have an increased ability to launch TBMs. It is also important to note that there is still an increasing marginal return for the addition of more TEL assets even with a sensor network. This again reenforces the notion that one response any nation has to remain a threat even in the face of advancing technology is to obtain larger numbers of forces.

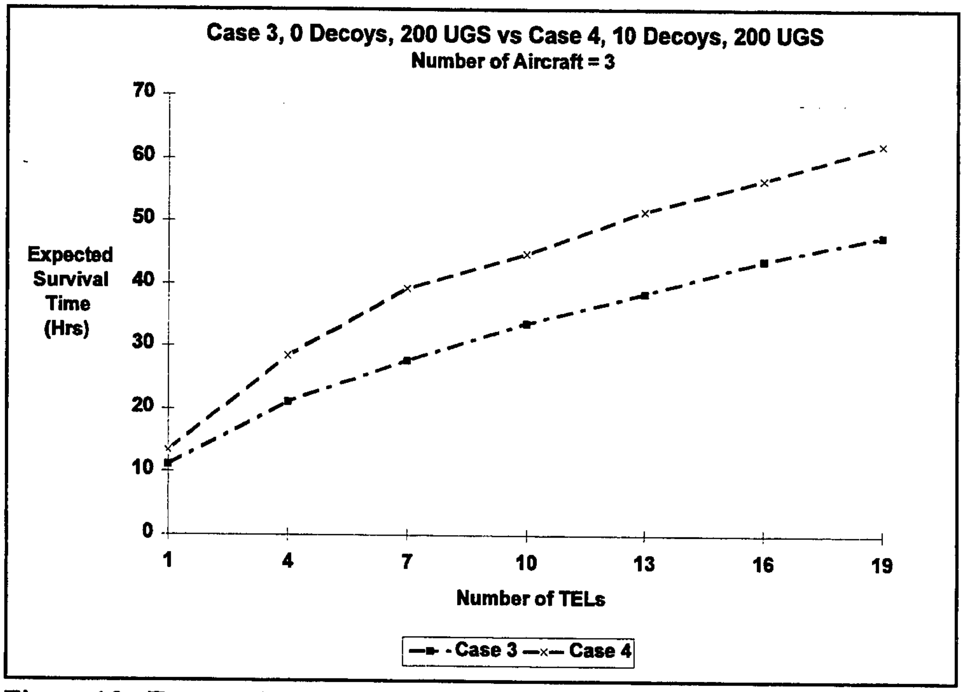

Figure 18. Expected Survival Time, Case 3 versus Case 4.

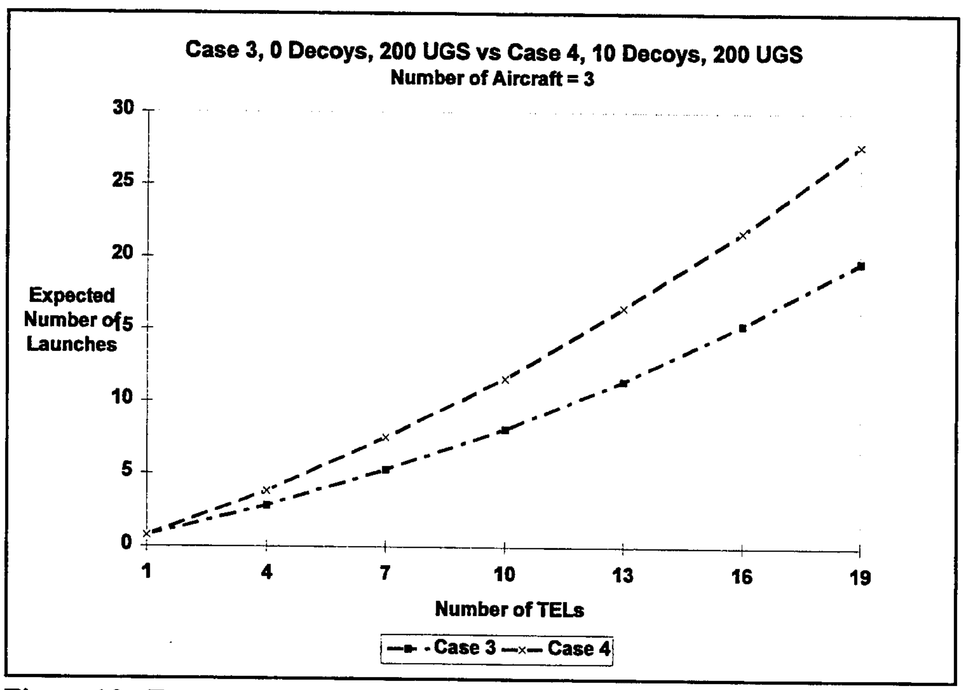

Figure 19. Expected Number of Launches, Case 3 versus Case 4.

3. Case 1 versus Case 3

Although the previous discussion seems to downplay the importance of having a UGS network, the comparison of Case 1 to Case 3 shows the potential of such a system. Figure 20 provides the simulation results for the expected survival time. The 200 UGS system locations provide only a 2.5 percent increase in the area covered but reduce the survival time by about 50 percent in the case of 19 TELs. This lends credence to the acquisition and the use of UGSs in support of attack operations. Figure 21 is a graph of the expected number of launches. Again the UGS network has reduced the number of TBMs launched in theater. The functional form of the expected number of launch curve still exhibits an increasing marginal return for the addition of TELs in the Op Area.

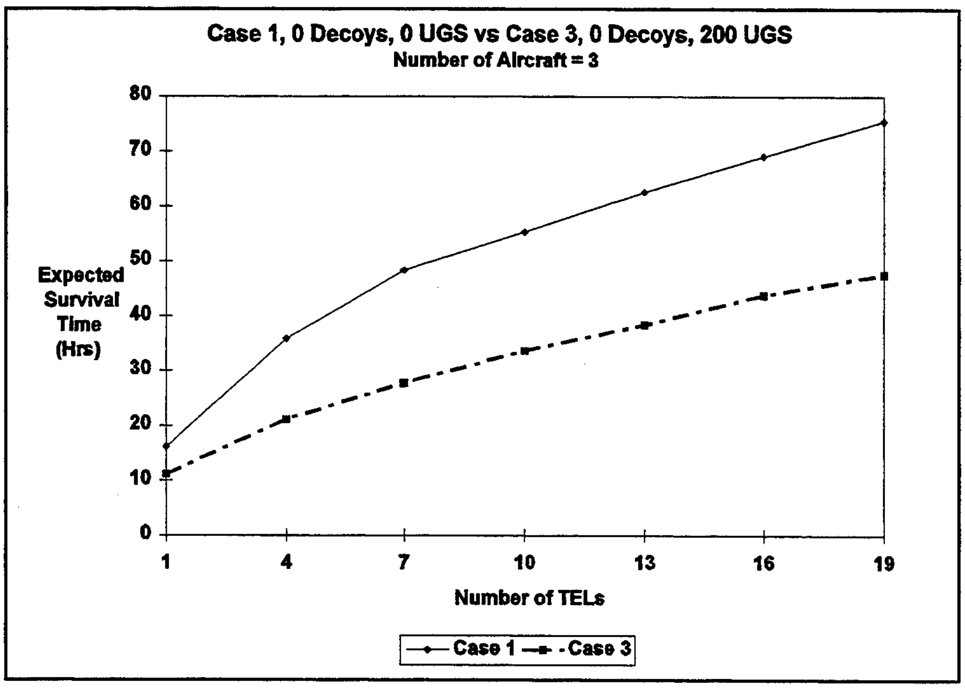

Figure 20. Expected Survival Time, Case 1 versus Case 3.

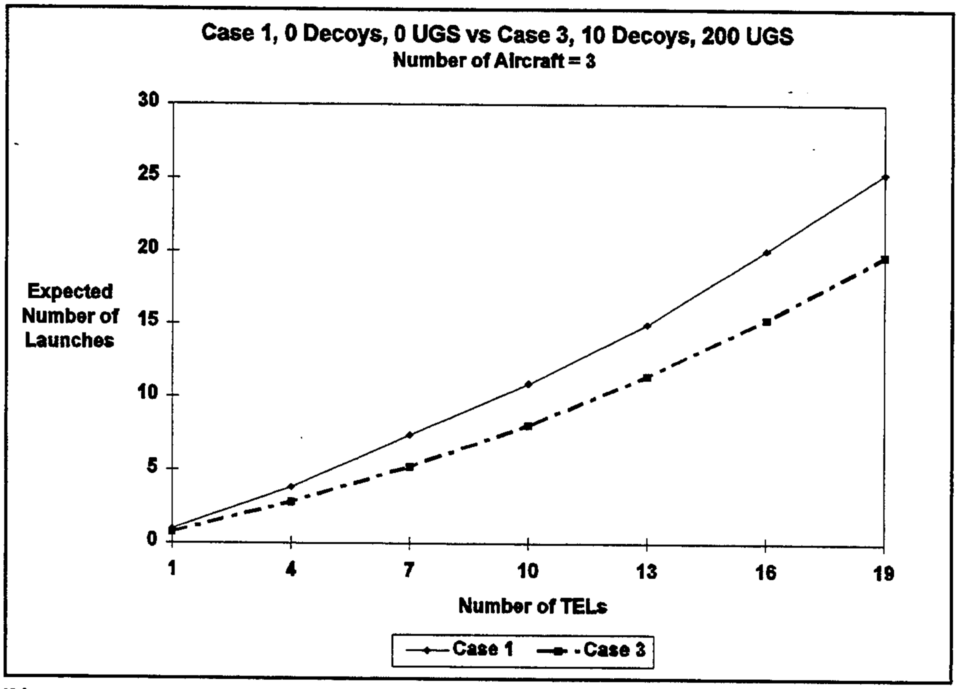

Figure 21. Expected Number of Launches, Case 1 versus Case 3.

4. Case 2 Versus Case 4

The final direct comparison is brought forward for additional emphasis on the themes highlighted previously. Figures 22 and 23 display the results of the simulation for Case 2, in which decoy TELs are present without a UGS network, and Case 4, where both decoy TELs and a 200-location UGS network are included. The presence of the UGS network has the same effect as before: a reduction in survival time and number of launches, but the curves have the same functional form.

5. Inclusive Comparison

Figures 24 and 25 incorporate the results for each MOE. The results of all four Cases are presented together to summarize the general results discussed in this section. The benefit of deception for TEL forces is easily seen, as is the effect of a UGS network capability.

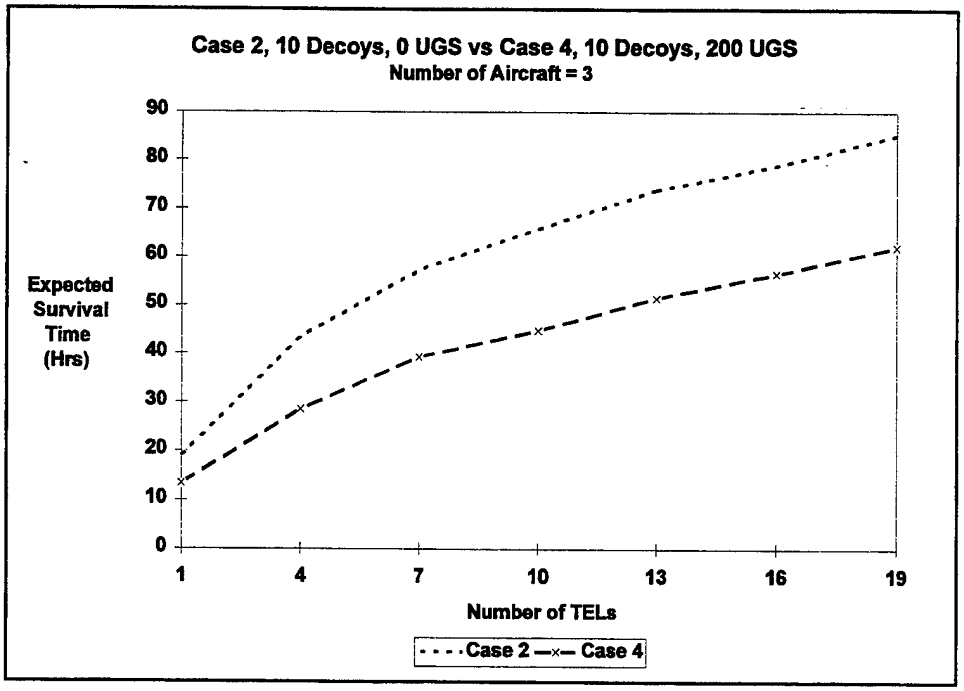

Figure 22. Expected Survival Time, Case 2 versus Case 4.

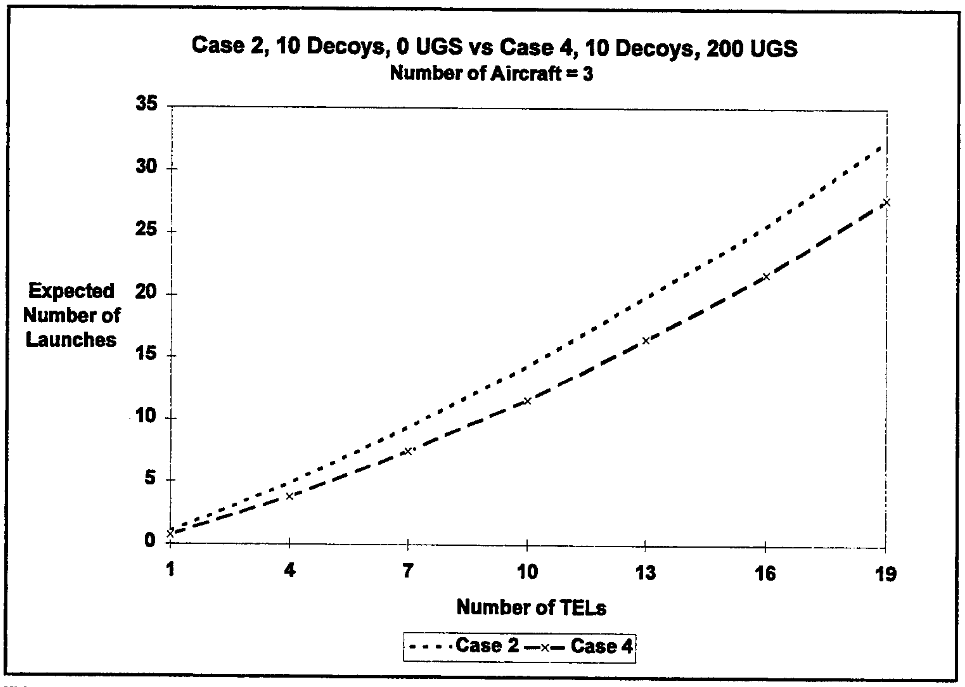

Figure 23. Expected Number of Launches, Case 2 versus Case 4.

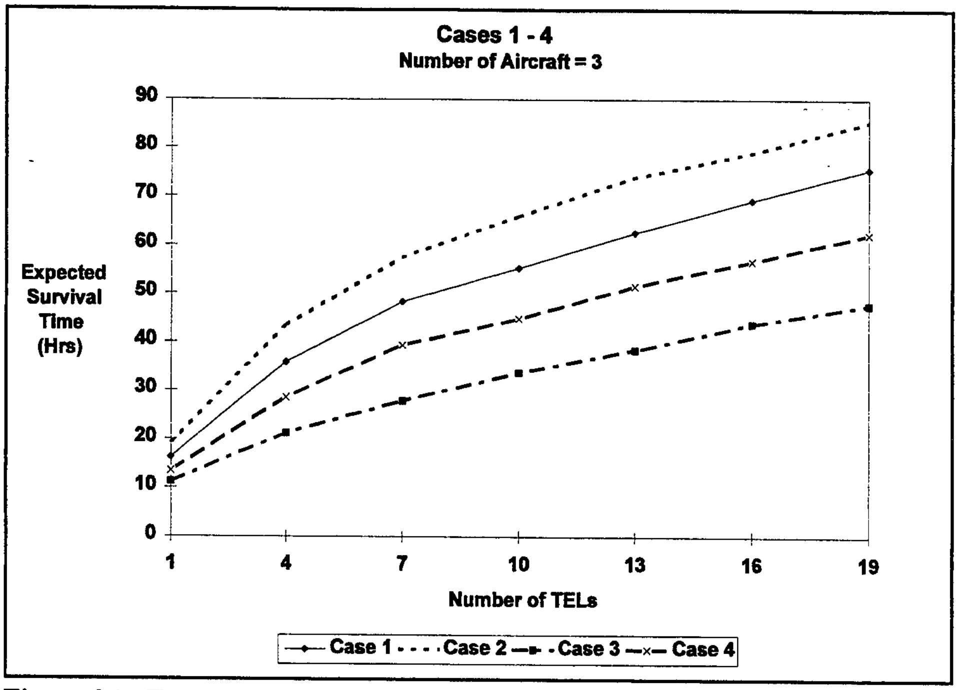

Figure 24. Expected Survival Time, Cases 1-4.

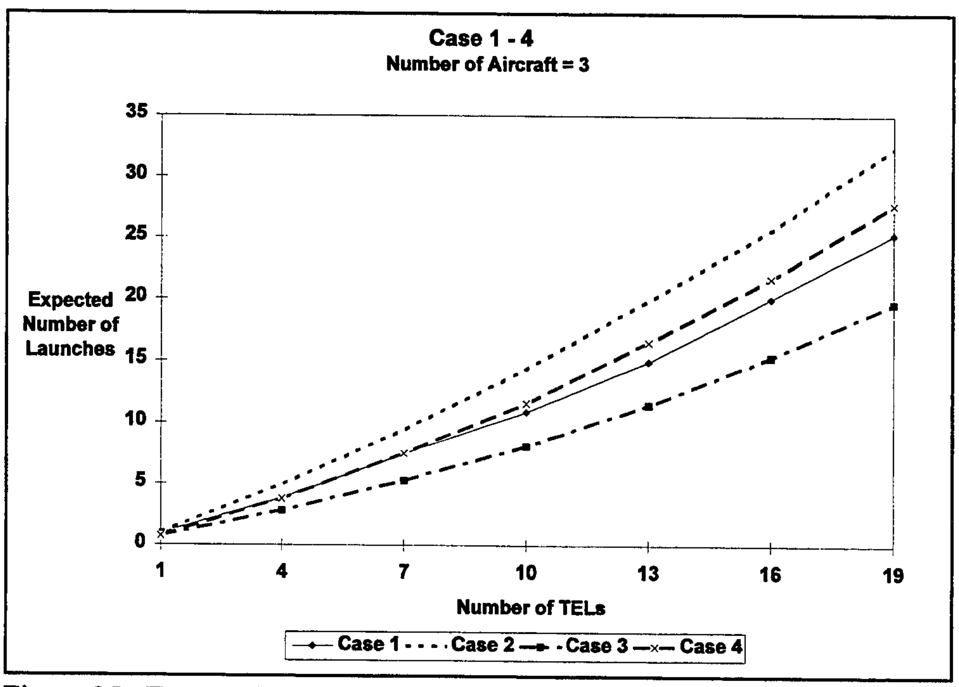

Figure 25. Expected Number of Launches, Case 1 - 4.

V. CONCLUSION

A. SUMMARY

This thesis attempted to shed light on the effects that certain products of a revolution in military affairs would have upon both U.S. forces and the forces of any potential adversary. The methodology for this study began with the development of a wartime scenario based on attack operations. A simulation was constructed in accordance with a unique set of stylized assumptions. The simulation assumed technological improvements to TEL employment capabilities and applied the concept of dominant battlespace awareness to U.S. forces through the introduction of UGS assets. The results of the simulation are obviously dependent on the simulation formulation itself, and thus are not presented as predictions, but have been used to highlight several important trends.

The predominant assertion of most writings on the subject of revolutions in military affairs that this author has reviewed tends to focus on the benefits of promoting this revolution to U.S. forces. I also believe that this same revolution holds promise for other nations, and that to ignore this fact may have consequences yet unseen. Although dependent on the stylized assumptions and parameters of the simulation, the TBM campaigns could be loosely judged a success. The advanced support network required for today's TELs to operate in the same fashion is not an unrealistic capability.

Deception techniques have always been used in warfare. As the U.S. commits to more and more weapons using increased levels of technology, those concerned with military

affairs should screen pronouncements of capability against potential deception techniques. Although not specifically addressed in this thesis, deception by means of new technologies may provide our opponents with opportunity for success in war.

Another conclusion which may be drawn from this thesis supports the precept of dominant battlespace awareness. The unattended ground sensor system modeled in the simulation provides an enhanced capability to conduct attack operations. The general performance of the simulation in this regard is certainly expected. The UGS network, although stationary, is an addition of sensors at no "cost" to the simulation. That is, the sensors provided extra information, in real time, to aircraft which were using the same search methodology whether sensors were there or not.

Finally, although not precisely quantified, the simulation behaved in accordance with another general rule of warfare: force ratios matter. As the number of TELs increased, the TEL forces were able to launch TBMs at an increasing rate. The addition of more aircraft provided the only counter to this trend. The notion that even a technologically advanced system will suffer when faced with large numbers of targets certainly seems intuitive.

B. ADDITIONAL RESEARCH RECOMMENDATIONS

There are many additional areas for further research suggested by this thesis. Those items introduced previously are reiterated here for additional emphasis:

- Investigating improved search tactics for attack aircraft supported by unattended ground sensors.
- Exploring the effect of bomb damage assessment on attack operations.
- Analysis of tradeoffs between attack operation capabilities and force sizes.

There are several more themes worth mentioning. The first is that the basic scenario of attack aircraft searching for TELs was simulated in a stylized fashion. Any of the core assumptions (the modeling of UGS networks as a "cookie cutter" detector, for example) are open avenues for exploration. No model captures all the elements of reality. Decisions must be made as to the essential elements. Research into the criteria that are important to the modeling of a given warfare scenario certainly serve as a source of a follow-on topic.

There are many new technologies that the United States will incorporate into its warfare capabilities in the future. This thesis explored the effect a doctrine of deception had upon a system dependent on a technological capability. Any system of warfare has an Achilles heel. Timely research exposing a future system's potential weaknesses has merit.

APPENDIX A. PROLIFERATION OF TBMS

This appendix lists the countries that are known to possess tactical ballistic missiles systems. The information contained here is unclassified and from open sources.

	County	System
1	Afghanistan	SCUD-B
2	Algeria	SCUD-B, FROG-7
3	Argentina	Alacran, CONDOR 2
4	Azerbaijan	SCUD-B
5	Belarus	SS-21, SCUD-B
6	Belgium	LANCE
7	Brazil	MB/EE-150,-350,-600,-100, SS-300, SS-100
8	Bulgaria	SCUD-B
9	Czech Rep.	SS-21, SCUD -B
10	Egypt	SCUD-B, VECTOR, SAKE 80, SCUD-B, CONDOR 2
11	France	PLUTON, HADES
12	Georgia	SCUD-B
13	Germany	LANCE
14	Hungary	SS-21, SCUD-B
15	Iran	SCUD-B, M-18, EAGLE
16	Iraq	AL HUSAYN, ALL ABBAS, SCUD-B, SS-12, SS-21
17	Israel	JERICHO I & II, LANCE
18	Italy	LANCE
19	Kazakhstan	SS-21, SCUD-B

20	Kuwait	FROG-7
21	Libya	SS-21, SCUD-B, AL FATAH, LAYTH, FROG-7
22	Netherlands	LANCE
23	PRC	M-7, M-9, M-11, CSS-2, CSS-1, CSS-5, DF-25
24	PRK	NO DONG I, SCUD-B
25	Pakistan	HAIFT I, II, & 3
26	Poland	SS-21, SCUD-B
27	Romania	SCUD-B
28	Russia	SS-21, SCUDS A-D, FROG-7, SS-4, SS-12, SS-23
29	S. Africa	ARNISTON
30	S. Korea	KOR SSM (NHK-1)
31	Saudi Arabia	CSS-2
32	Slovakia	SS-21, SCUD-B
33	Syria	SS-21, SCUD-B
34	UK	LANCE
35	USA	LANCE, ATACMS
36	Ukraine	SS-21, SCUD-B
37	Vietnam	SCUD-B
38	Yemen	SS-21, SCUD-B, FROG-7

APPENDIX B. COMPARISON CASE RESULTS

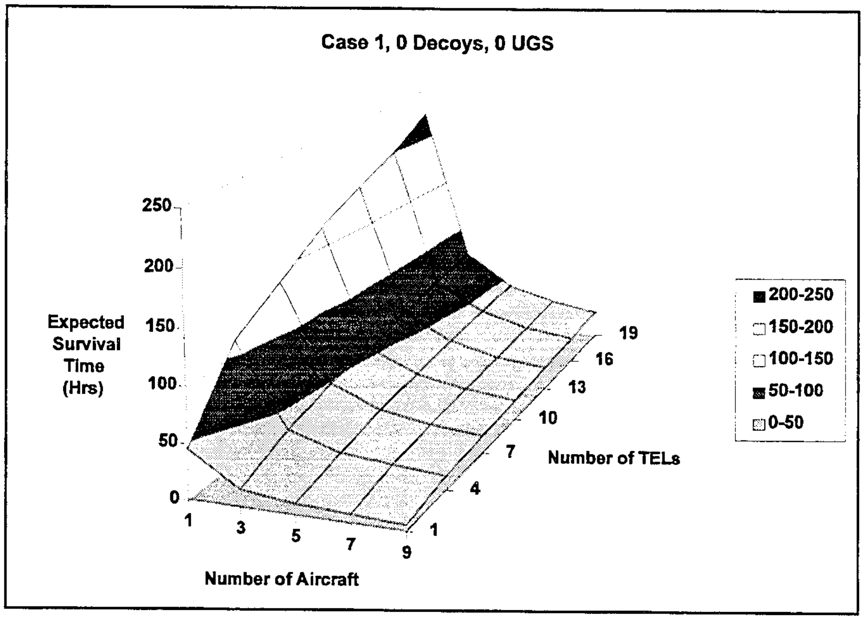

Figure 26. Three-Dimensional Surface Plot of the Expected Survival Time for Case 1.

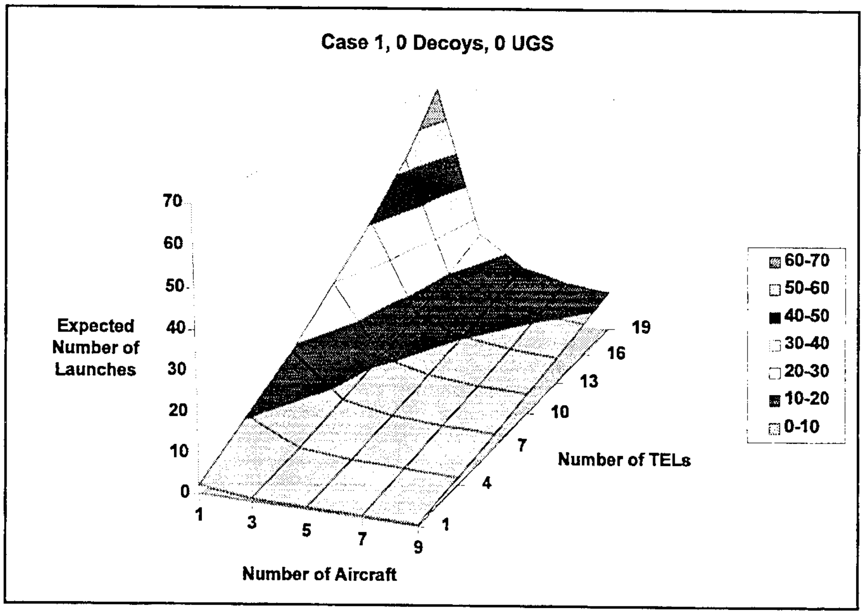

Figure 27. Three-Dimensional Surface Plot of the Expected Number of Launches for Case 1.

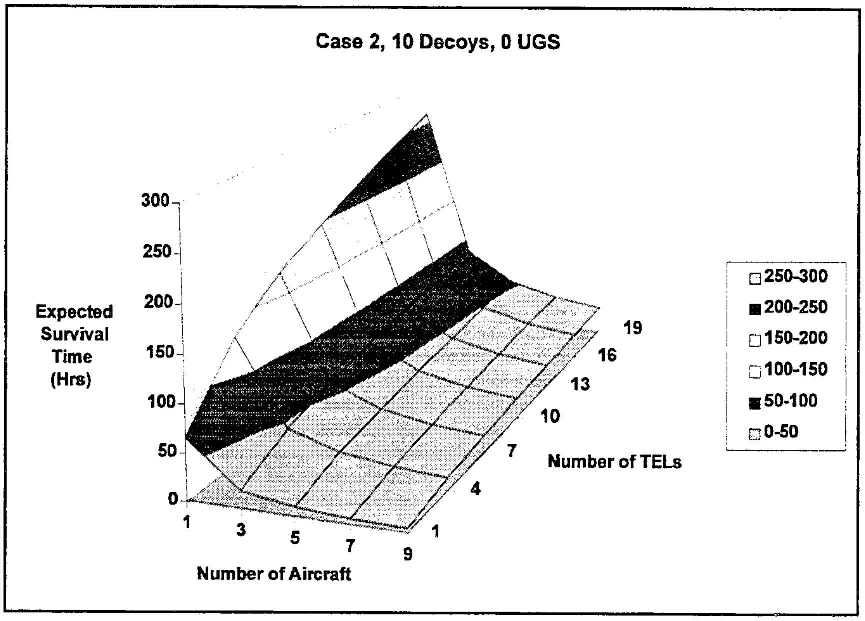

Figure 28. Three-Dimensional Surface Plot of the Expected Survival Time for Case 2.

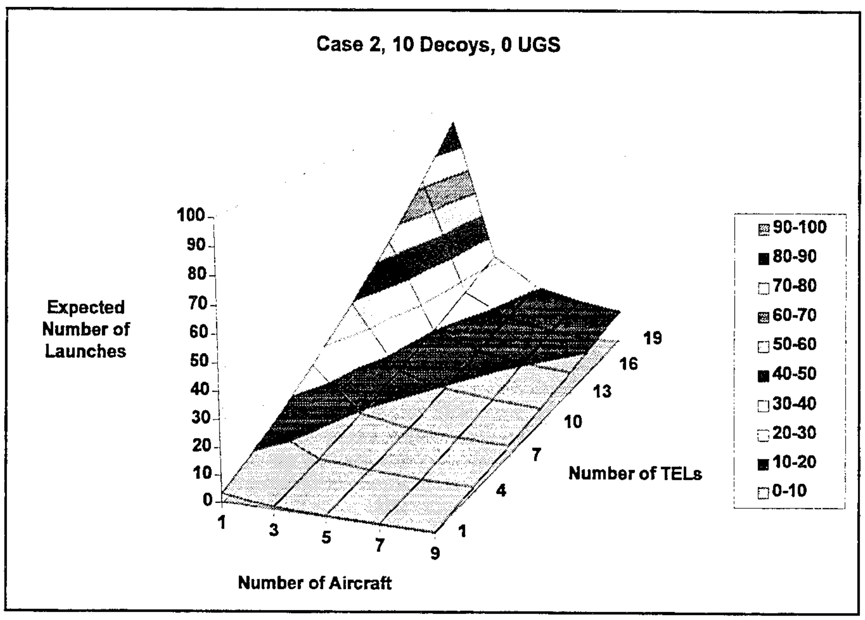

Figure 29. Three-Dimensional Surface Plot of the Expected Number of Launches for Case 2.

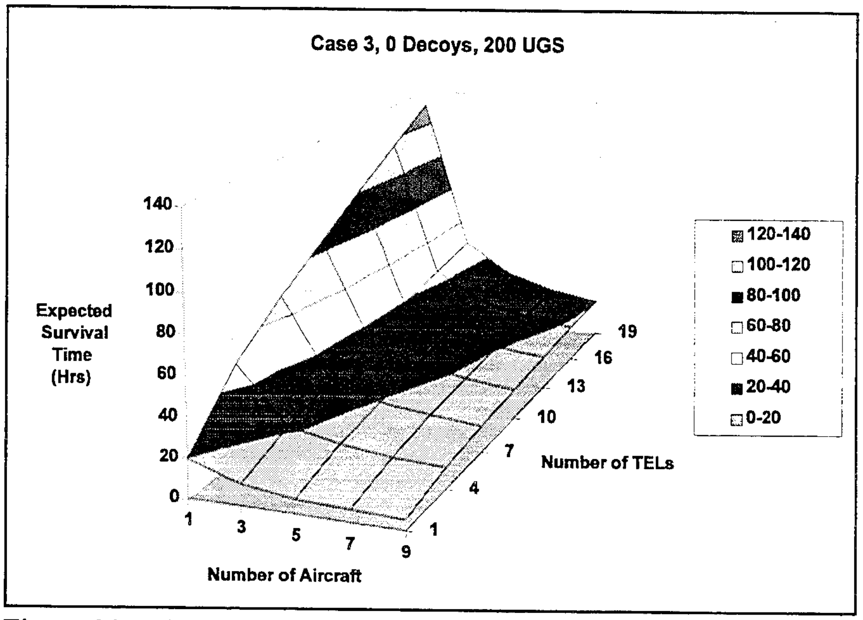

Figure 30. Three-Dimensional Surface Plot of the Expected Survival Time for Case 3.

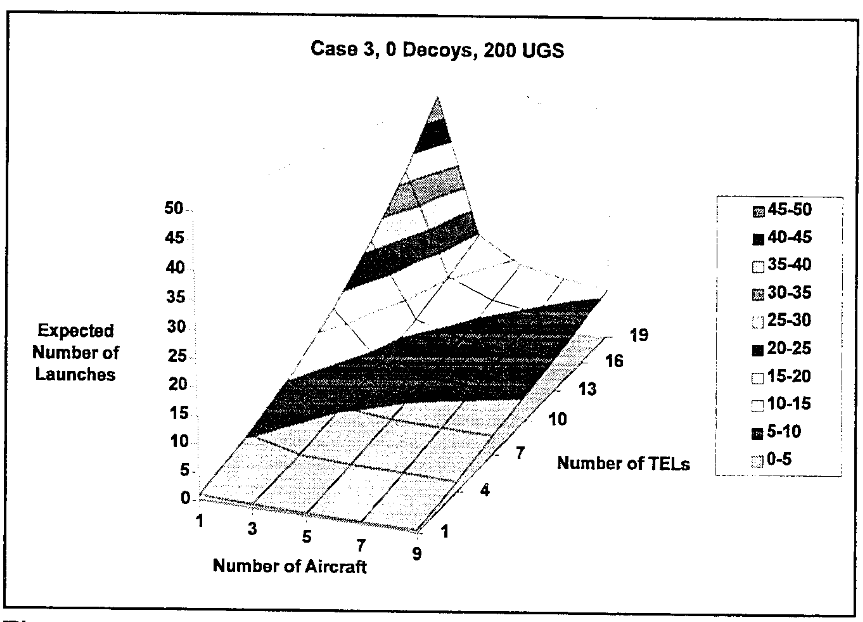

Figure 31. Three-Dimensional Surface Plot of the Expected Number of Launches for Case 3.

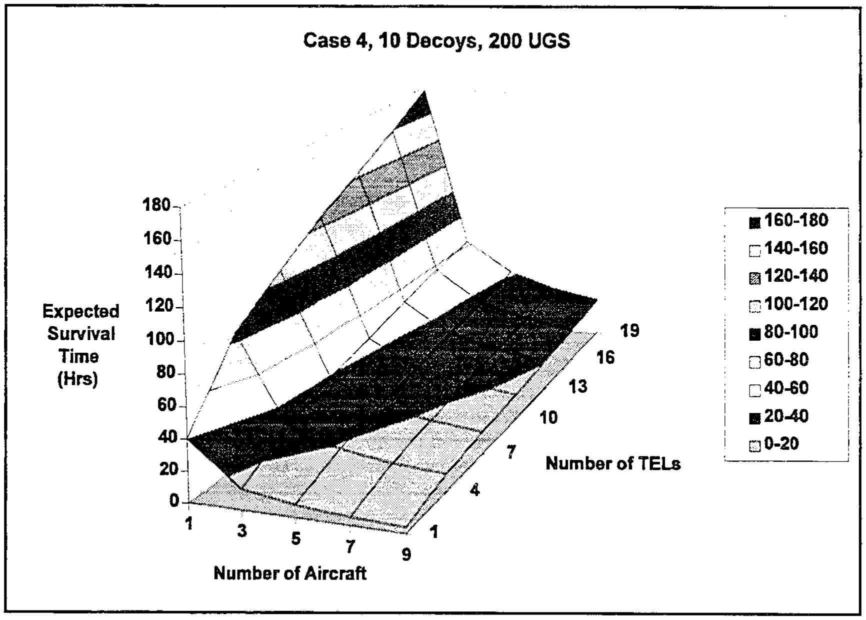

Figure 32. Three-Dimensional Surface Plot of the Expected Survival Time for Case 4.

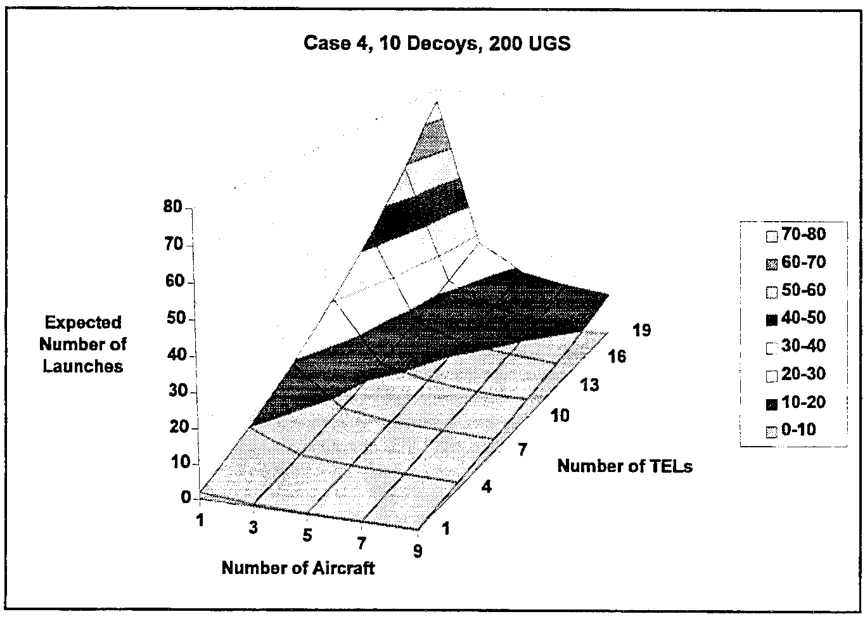

Figure 33. Three-Dimensional Surface Plot of the Expected Number of Launches for Case 4.

APPENDIX C. SIMULATION PROGRAM

This appendix contains the computer program used for the simulation. The simulation was written in MODSIM II®, a modular, object-oriented simulating language. Each of the modules is presented as a separate section with the section title reflecting the module name as required by the MODSIM II® compiler. Definition modules precede the corresponding implementation modules.

A. MATTACK.MOD

```
MAIN MODULE Attack;

FROM System IMPORT System;
FROM SimMod IMPORT StartSimulation;
FROM UtilMod IMPORT DateTime;

VAR
  TELs, Air, Decoys, UGS, Runs : INTEGER;
  start,end : STRING;

BEGIN
  DateTime (start);
  INPUT (Decoys);
  INPUT (UGS);
  NEW (System);
  FOR TELs:= 1 TO 19 BY 3
    FOR Air:= 1 TO 9 BY 2
      ASK System TO SetParameters(TELs,Decoys,Air,UGS);
      FOR Runs := 1 TO System.NumberRuns
        TELL System TO StartCampaign;
        StartSimulation;
        ASK System TO ResetCampaign;
      END FOR;
      ASK System TO OutputReport;
      ASK System TO Reset;
    END FOR;
  END FOR;
  ASK System TO EchoInput;
  DISPOSE (System);
  DateTime (end);
  OUTPUT (start);
```

```
  OUTPUT (end);
END MODULE.
```

B. DMOVING.MOD

```
DEFINITION MODULE Moving;

TYPE
  PointType = RECORD
    Xcoordinate , YCoordinate   :REAL;
  END RECORD;

  PathObj = OBJECT
    Time : REAL;
    Where : PointType;
    ASK METHOD ObjInit;
    ASK METHOD ObjTerminate;
    ASK METHOD GetPoint (IN T : REAL; IN Posit : PointType);
    ASK METHOD PrintPoint;
  END OBJECT;

  MovingObj = OBJECT
    CurrentPosition , Destination : PointType;
    Velocity , StartTime : REAL;
    ASK  METHOD ObjInit;
    ASK  METHOD SetPosition (IN Location : PointType);
    ASK  METHOD SetDestination (IN Location : PointType);
    ASK  METHOD Distance (IN P1,P2:PointType; OUT D: REAL);
    ASK  METHOD ChangeVelocity (IN NewVelocity : REAL);
    ASK  METHOD CalcPosit(IN Time : REAL; INOUT Posit : PointType);
  END OBJECT;

END MODULE.
```

C. IMOVING.MOD

```
IMPLEMENTATION MODULE Moving;

FROM MathMod IMPORT SQRT;
FROM SimMod IMPORT SimTime;
FROM OpArea IMPORT OpArea;

OBJECT PathObj;

ASK METHOD ObjInit;
BEGIN
  NEW(Where);
END METHOD;

ASK METHOD ObjTerminate;
```

```
BEGIN
   DISPOSE(Where);
END METHOD;

ASK METHOD GetPoint (IN T:REAL; IN Posit : PointType);
BEGIN
   Time := T;
   Where.XCoordinate := Posit.XCoordinate;
   Where.YCoordinate := Posit.YCoordinate;
END METHOD;

ASK METHOD PrintPoint;
BEGIN
   OUTPUT (Time," ",Where.XCoordinate," ",Where.YCoordinate);
END METHOD;

END OBJECT;

OBJECT MovingObj;

ASK METHOD ObjInit;
BEGIN
   NEW (CurrentPosition);
   NEW (Destination);
END METHOD;

ASK METHOD SetPosition (IN Location : PointType);
BEGIN
   CurrentPosition.XCoordinate := Location.XCoordinate;
   CurrentPosition.YCoordinate := Location.YCoordinate;
END METHOD;
ASK METHOD SetDestination (IN Location:PointType);
BEGIN
   Destination.XCoordinate:=Location.XCoordinate;
   Destination.YCoordinate:=Location.YCoordinate;
END METHOD;

ASK METHOD Distance (IN P1,P2 : PointType; OUT D : REAL);
VAR
   X,Y :REAL;

BEGIN
   X := P1.XCoordinate - P2.XCoordinate;
   Y := P1.YCoordinate - P2.YCoordinate;
   D := SQRT(X*X+Y*Y);
END METHOD;

ASK METHOD ChangeVelocity (IN NewVelocity : REAL);
BEGIN
   Velocity := NewVelocity;
END METHOD;
```

```
ASK  METHOD CalcPosit (IN Time : REAL; INOUT Posit : PointType) ;
VAR
  TimePassage, DeltaX, DeltaY, D : REAL;
BEGIN
  TimePassage:= Time - StartTime;
  IF TimePassage <> 0.0
  IF    (Destination.XCoordinate <> CurrentPosition.XCoordinate) AND
        (Destination.YCoordinate <> CurrentPosition.YCoordinate)
    DeltaX := Destination.XCoordinate - CurrentPosition.XCoordinate;
    ASK SELF Distance (CurrentPosition,Destination,D);
    DeltaX := DeltaX / D;
    DeltaX := DeltaX * Velocity * TimePassage;
    Posit.XCoordinate := CurrentPosition.XCoordinate + DeltaX;
    DeltaY := Destination.YCoordinate - CurrentPosition.YCoordinate;
    DeltaY := DeltaY / D;
    DeltaY := DeltaY * Velocity*TimePassage;
    Posit.YCoordinate := CurrentPosition.YCoordinate + DeltaY;
  ELSE
    Posit.XCoordinate := CurrentPosition.XCoordinate;
    Posit.YCoordinate := CurrentPosition.YCoordinate;
  END IF;
  ELSE
    Posit.XCoordinate := CurrentPosition.XCoordinate;
    Posit.YCoordinate := CurrentPosition.YCoordinate;
  END IF;
END METHOD;

END OBJECT;

END MODULE.
```

D. DSIMPLESTATS.MOD

```
DEFINITION MODULE SimpleStats ;

{This module and the accompanying implementation module were first constructed as part of an assignment
during OA3302, Modeling and Simulation. As part of the thesis the author's original work was modified to
adhere to the assignment solution guide.}

TYPE
  StatObj = OBJECT
    Sum, SumOfSquares, Min, Max : REAL;
    N : INTEGER;
    ASK METHOD ObjInit;
    ASK METHOD GetSample (IN X : REAL);  {Adds a new observation}
    ASK METHOD Mean() : REAL;              {Returns the mean of the sample}
    ASK METHOD Variance() : REAL;            {Returns the Variance of the sample}
    ASK METHOD StdDev() : REAL;              {Returns the Std. Dev. of the sample}
    ASK METHOD CIHalfWidth() : REAL;       {Gives half-width of CI}
    ASK METHOD Output;                       {Output report of all statistics}
    ASK METHOD Reset;                        {Removes observations}
```

```
  END OBJECT;

END MODULE.
```

E. ISIMPLESTATS.MOD

```
IMPLEMENTATION MODULE SimpleStats ;

FROM MathMod IMPORT SQRT;


OBJECT StatObj;

ASK METHOD ObjInit;
BEGIN
  Reset;
END METHOD;

ASK METHOD GetSample(IN X: REAL); {Adds a new observation}
BEGIN
  Sum := Sum + X ;
  SumOfSquares := SumOfSquares + X * X ;
  Min := MINOF (Min, X);
  Max := MAXOF (Max, X);
  N := N + 1;
END METHOD;

ASK METHOD Mean () : REAL;  {Returns the mean of the sample}
BEGIN
  RETURN ( Sum / FLOAT(N) );
END METHOD;

ASK METHOD Variance() : REAL;  {Returns the Variance of the sample}
BEGIN
  IF N > 1
    RETURN ( (SumOfSquares - Sum * Sum / FLOAT(N) )/ FLOAT(N-1) );
  ELSE
    RETURN 0.0 ;
  END IF;
END METHOD;

ASK METHOD StdDev() : REAL; {Returns the Std. Dev. of the sample}
BEGIN
  RETURN SQRT( Variance() );
END METHOD;

ASK METHOD CIHalfWidth() : REAL;      {Gives width of CI}
BEGIN
  IF N > 1
    RETURN SQRT(Variance() / FLOAT(N)) * 1.959963984540053 ;
```

```
  ELSE
    RETURN MAX(REAL);
  END IF;
END METHOD;

ASK METHOD Output; {Output report of all statistics}
VAR
  theLower, theUpper : REAL;
BEGIN
  OUTPUT (" N   Min    Max    Mean   Variance   CI Halfwidth");
  OUTPUT ("-----------------------------------------------");
  OUTPUT (N, " ", Min, " ", Max, " ", Mean(), " ", Variance(), " ", CIHalfWidth() );
END METHOD;

ASK METHOD Reset; {Removes observations}
BEGIN
  Sum := 0.0;
  SumOfSquares := 0.0;
  Min := MAX(REAL);
  Max := MIN(REAL);
  N := 0;
END METHOD;

END OBJECT;

END MODULE.
```

F. DSYSTEM.MOD

```
DEFINITION MODULE System;

FROM SimpleStats IMPORT StatObj;

TYPE
  SystemObj = OBJECT
    NumberRuns : INTEGER;
    PreHostile : REAL;
    CampaignTime, NumberLaunches, TimeHidden, TimeMoving : StatObj;
    ASK METHOD ObjInit;
    ASK METHOD EchoInput;
    ASK METHOD SetParameters(IN TELs ,Decoys, Air, UGS : INTEGER);
    TELL METHOD StartCampaign;
    ASK METHOD ResetCampaign;
    ASK METHOD Reset;
  ASK METHOD OutputReport;
  ASK METHOD ObjTerminate;
END OBJECT;

VAR
```

```
   System : SystemObj;

END MODULE.
```

G. ISYSTEM.MOD

```
IMPLEMENTATION MODULE System;

FROM OpArea IMPORT OpArea;
FROM SimMod IMPORT SimTime, ResetSimTime;
FROM BlueForce IMPORT SquadronCO;

OBJECT SystemObj;

ASK METHOD ObjInit;
BEGIN
   NEW (CampaignTime);
   NEW (NumberLaunches);
   NEW (TimeHidden);
   NEW (TimeMoving);
   INPUT (NumberRuns);
   INPUT (PreHostile);
   PreHostile := PreHostile * 24.0; {Number of days -> hours }
   NEW (OpArea);
   NEW (SquadronCO);
END METHOD;

ASK METHOD EchoInput;
BEGIN
   OUTPUT ();
   OUTPUT ("Number of Runs = ",NumberRuns);
   OUTPUT ("Days in PreHostility Phase = ",TRUNC(PreHostile/24.0));
   ASK OpArea TO EchoInput;
   ASK SquadronCO TO EchoInput;
   OUTPUT ();
   OUTPUT ("Expected Time Hidden = ",ASK TimeHidden Mean());
   OUTPUT ("Expected Time Moving = ",ASK TimeMoving Mean());
END METHOD;

ASK METHOD SetParameters(IN TELs, Decoys, Air, UGS : INTEGER);
BEGIN
   ASK OpArea TO SetParameters(TELs, Decoys);
   ASK SquadronCO TO SetParameters(Air, UGS);
END METHOD;

TELL METHOD StartCampaign;
BEGIN
   ASK OpArea TO GenerateTargets;
   WAIT DURATION PreHostile;
   END WAIT;
```

```
   ASK SquadronCO TO StartSearching;
END METHOD;

ASK METHOD ResetCampaign; {This is for a reset between campaign runs}
BEGIN
   ResetSimTime(0.0);
   ASK OpArea TO ResetCampaign;
   ASK SquadronCO TO ResetCampaign;
END METHOD;

ASK METHOD Reset; {This is for a reset before a change of parameters}
BEGIN
   ASK CampaignTime TO Reset;
   ASK NumberLaunches TO Reset;
   ASK OpArea TO Reset;
   ASK SquadronCO TO Reset;
END METHOD;

ASK METHOD OutputReport;
BEGIN
   OUTPUT (ASK OpArea NumberTELs," ",ASK OpArea NumberDecoys," ",
   ASK SquadronCO NumberAircraft," ",ASK SquadronCO NumberSensors," ",
   ASK CampaignTime Mean()," ",ASK CampaignTime Variance()," ",
   ASK CampaignTime CIHalfWidth()," ", ASK NumberLaunches Mean()," ",
   ASK NumberLaunches Variance()," ",ASK NumberLaunches CIHalfWidth());
END METHOD;

ASK METHOD ObjTerminate;
BEGIN
   DISPOSE (CampaignTime);
   DISPOSE (NumberLaunches);
   DISPOSE (TimeHidden);
   DISPOSE (TimeMoving);
   DISPOSE (OpArea);
   DISPOSE (SquadronCO);
END METHOD;

END OBJECT;

END MODULE.
```

H. DOPAREA.MOD

```
DEFINITION MODULE OpArea;

FROM RandMod IMPORT RandomObj;
FROM Moving IMPORT MovingObj, PointType;
FROM GrpMod IMPORT QueueObj;

TYPE
```

```
  UnitType = (TEL, Decoy);

  VehicleObj = OBJECT(MovingObj);
    Class : UnitType;
    SerialNumber : INTEGER;
    Alive, Hidden : BOOLEAN;
    ASK METHOD ObjTerminate;
    ASK METHOD SetParameters (IN Type : UnitType; IN Num1: INTEGER);
    ASK METHOD ID: INTEGER;
    TELL METHOD ConductMission;
    TELL METHOD MoveTo (IN There : PointType);
    ASK METHOD Explode(IN Time:REAL);
    ASK METHOD Reset;
  OVERRIDE
    ASK METHOD ObjInit;
  END OBJECT;

  RegionObj = OBJECT;
    NumberTELs, NumberDecoys, TargetsKilled, Launch, NumberBases,
    MinNumberBases, BaseFactor : INTEGER;
    Xcoord, Ycoord, WaitingTime, ReloadTime, RandomSpeed,
    RandomBase : RandomObj;
    VehicleList : QueueObj;
    Hostile : BOOLEAN;
    BaseList : ARRAY INTEGER OF PointType;
    AveSpeed, EvasionSpeed, LoWaitTime, MeanWaitTime, HiWaitTime,
    LoReloadTime, MeanReloadTime,  HiReloadTime,
    MinX, MaxX, MinY, MaxY : REAL;
    ASK METHOD ObjInit;
    ASK METHOD ObjTerminate;
    ASK METHOD SetParameters (IN TELs, Decoys : INTEGER);
    ASK METHOD EchoInput;
    ASK METHOD MakeBaseList (IN Num1 : INTEGER);
    ASK METHOD PickBase(INOUT Location:PointType);
    ASK METHOD GetCoordinate(INOUT Location :PointType);
    ASK METHOD PickSpeed (IN Mean: REAL) : REAL;
    ASK METHOD DestroyTarget (IN Time:REAL; IN Target:VehicleObj);
    ASK METHOD GenerateTargets;
    ASK METHOD CountLaunch;
    ASK METHOD ResetCampaign;
    ASK METHOD Reset;
  END OBJECT;

VAR
  OpArea: RegionObj;

END MODULE;
```

I. IOPAREA.MOD

```
IMPLEMENTATION MODULE OpArea;

FROM Moving IMPORT PointType;
FROM SimMod IMPORT SimTime, Interrupt;
FROM System IMPORT System;
FROM BlueForce IMPORT SquadronCO;
FROM RandMod IMPORT FetchSeed;

OBJECT VehicleObj;

ASK METHOD ObjInit;
BEGIN
   NEW (CurrentPosition);
   NEW (Destination);
   Alive := TRUE;
   Hidden := FALSE;
END METHOD;

ASK METHOD Reset;
BEGIN
   Alive:=TRUE;
   Hidden:=FALSE;
END METHOD;

ASK METHOD ObjTerminate;
VAR
   Count : INTEGER;
BEGIN
   DISPOSE (CurrentPosition);
   DISPOSE (Destination);
END METHOD;

ASK METHOD SetParameters (IN Type : UnitType; IN Num1 : INTEGER);
BEGIN
   Class:=Type;
   SerialNumber:=Num1;
END METHOD;

ASK METHOD ID : INTEGER;
BEGIN
   RETURN SerialNumber;
END METHOD;

TELL METHOD ConductMission;
VAR
   D : REAL;
   Location : PointType;
BEGIN
   IF Alive
      StartTime := SimTime();
      NEW (Location);
```

```
ASK  SELF TO ChangeVelocity
   (ASK OpArea TO PickSpeed (ASK OpArea AveSpeed));
ASK OpArea TO GetCoordinate (Location);
ASK SELF TO SetDestination (Location);
IF SimTime() > ASK System PreHostile
   ASK SquadronCO TO DetermineDetections (SELF);
END IF;
ASK SELF Distance (CurrentPosition, Destination, D);
WAIT DURATION (D / Velocity);
   ON INTERRUPT
   DISPOSE (Location);
      ASK System.TimeMoving TO GetSample (SimTime() - StartTime);
   TERMINATE;
END WAIT;
ASK System.TimeMoving TO GetSample (SimTime ) - StartTime);
ASK SELF TO SetPosition (Location);
Hidden :=TRUE;
ASK SquadronCO TO StopDetection (ASK SELF SerialNumber);
StartTime:=SimTime();
WAIT DURATION ASK OpArea.WaitingTime Triangular
   (ASK OpArea LoWaitTime, ASK OpArea MeanWaitTime,
   ASK OpArea HiWaitTime);
END WAIT;
Hidden:= FALSE;
ASK System.TimeHidden TO GetSample (SimTime() - StartTime);
IF SimTime() > ASK System PreHostile
   ASK OpArea TO CountLaunch;
END IF;
ASK OpArea TO PickBase (Location);
ASK SELF TO SetDestination (Location);
ASK SELF TO ChangeVelocity
   (ASK OpArea TO PickSpeed (ASK OpArea EvasionSpeed));
IF SimTime() > ASK System PreHostile
   ASK SquadronCO TO DetermineDetections (SELF);
END IF;
ASK SELF Distance (CurrentPosition,Destination, D);
StartTime := SimTime();
WAIT DURATION (D / Velocity);
   ON INTERRUPT
      DISPOSE (Location);
      ASK System.TimeMoving TO GetSample (SimTime() - StartTime);
   TERMINATE;
END WAIT;
ASK System.TimeMoving TO GetSample (SimTime() - StartTime);
Hidden:=TRUE;
ASK SELF TO SetPosition (Location);
ASK SquadronCO TO StopDetection (ASK SELF SerialNumber);
StartTime:=SimTime();
WAIT DURATION ASK OpArea.ReloadTime Triangular
   (ASK OpArea LoReloadTime, ASK OpArea MeanReloadTime,
   ASK OpArea HiReloadTime);
```

```
      END WAIT;
      ASK System.TimeHidden TO GetSample (SimTime() - StartTime);
      Hidden:=FALSE;
      DISPOSE (Location);
      TELL SELF TO ConductMission;
   END IF;
END METHOD;

TELL METHOD MoveTo (IN There : PointType);
VAR
   D : REAL;
   Location : PointType;
BEGIN
   IF (Alive) AND (ASK OpArea Hostile)
      StartTime := SimTime();
      NEW(Location);
      ASK SELF TO SetDestination (There);
      IF SimTime() > ASK System PreHostile
         ASK SquadronCO TO DetermineDetections (SELF);
      END IF;
      ASK SELF Distance (CurrentPosition, There, D);
      WAIT DURATION  (D / Velocity);
         ON INTERRUPT
            DISPOSE(Location);
         TERMINATE;
      END WAIT;
      ASK SELF TO SetPosition (There);
      ASK OpArea TO GetCoordinate (Location);
      ASK SELF TO SetDestination (Location);
      ASK SquadronCO TO StopDetection (ASK SELF SerialNumber);
      DISPOSE(Location);
      ASK SELF TO ChangeVelocity
         (ASK OpArea TO PickSpeed (ASK OpArea AveSpeed));
      TELL SELF TO MoveTo (Destination);
   END IF;
END METHOD;

ASK METHOD Explode (IN Time : REAL);
VAR
   Location : PointType;
BEGIN
   NEW (Location);
   CASE ASK SELF Class
      WHEN TEL:
         Interrupt (SELF, "ConductMission");
      OTHERWISE
         Interrupt (SELF, "MoveTo");
   END CASE;
   ASK SELF TO CalcPosit (Time, Location);
   Alive:=FALSE;
   ASK SELF TO SetPosition (Location);
```

```
    ASK SELF TO SetDestination (ASK SELF CurrentPosition);
    ASK SquadronCO TO StopDetection(ASK SELF SerialNumber);
    DISPOSE(Location);
END METHOD;

END OBJECT;

OBJECT RegionObj;

ASK METHOD ObjInit;
VAR
    Count, Number : INTEGER;
    Vehicle : VehicleObj;
    T : REAL;
BEGIN
    NEW (XCoord);
    NEW (YCoord);
    NEW (WaitingTime);
    NEW (ReloadTime);
    NEW (RandomSpeed);
    NEW (RandomBase);
    ASK XCoord TO SetSeed (FetchSeed (2));
    ASK YCoord TO SetSeed (FetchSeed (8));
    ASK WaitingTime TO SetSeed (FetchSeed (9));
    ASK ReloadTime TO SetSeed (FetchSeed (6));
    ASK RandomSpeed TO SetSeed (FetchSeed (7));
    ASK RandomBase TO SetSeed (FetchSeed (3));
    NEW (VehicleList);
    INPUT (AveSpeed);
    INPUT (EvasionSpeed);
    INPUT (LoWaitTime);
    INPUT (MeanWaitTime);
    INPUT (HiWaitTime);
    INPUT (LoReloadTime);
    INPUT (MeanReloadTime);
    INPUT (HiReloadTime);
    INPUT (MinNumberBases);
    INPUT (BaseFactor);
    INPUT (MinX);
    INPUT (MaxX);
    INPUT (MinY);
    INPUT (MaxY);
END METHOD;

ASK METHOD ObjTerminate;
VAR
    Vehicle : VehicleObj;
    Count : INTEGER;
BEGIN
    DISPOSE (XCoord);
    DISPOSE (YCoord);
```

```
  DISPOSE (WaitingTime);
  DISPOSE (RandomSpeed);
  DISPOSE (RandomBase);
  FOR Count := 1 TO ASK VehicleList numberIn
    Vehicle := ASK VehicleList TO Remove ();
    DISPOSE (Vehicle);
  END FOR;
  DISPOSE (VehicleList);
END METHOD;

ASK METHOD EchoInput;
BEGIN
  OUTPUT ("Number of Decoys = ", NumberDecoys);
  OUTPUT ();
  OUTPUT ("Normal Vehicle Speed = ", AveSpeed);
  OUTPUT ("Vehicle Evansion Speed = ", EvasionSpeed);
  OUTPUT ();
  OUTPUT ("Lower Bound for Mean Wait Time ", LoWaitTime);
  OUTPUT ("Mean Wait Time ", MeanWaitTime);
  OUTPUT ("Upper Bound for Mean Wait Time ", HiWaitTime);
  OUTPUT ();
  OUTPUT ("Lower Bound for Mean Reload Time ", LoReloadTime);
  OUTPUT ("Mean Reload Time ", MeanReloadTime);
  OUTPUT ("Upper Bound for Mean Reload Time ", HiReloadTime);
  OUTPUT ();
  OUTPUT ("Minimum Number of Bases = ", MinNumberBases);
  OUTPUT ("Logistic Base Factor = ", BaseFactor);
  OUTPUT ();
  OUTPUT ("X Range ", MinX, " ", MaxX);
  OUTPUT ("Y Range ", MinY, " ", MaxY);
END METHOD;

ASK METHOD SetParameters (IN TELs, Decoys:INTEGER);
VAR
  Vehicle : VehicleObj;
  Count, Total : INTEGER;
BEGIN
  Hostile := TRUE;
  TargetsKilled := 0;
  Launch := 0;
  Total := 0;
  NumberTELs := TELs;
  NumberDecoys := Decoys;
  ASK SELF TO MakeBaseList (NumberTELs);
  FOR Count := 1 TO NumberTELs
    NEW (Vehicle);
    ASK Vehicle TO SetParameters (TEL, Count);
    ASK VehicleList TO Add (Vehicle);
  END FOR;
  Total := NumberTELs;
  IF Decoys <> 0
```

```
      FOR Count := (Total + 1) TO (Total + NumberDecoys)
         NEW (Vehicle);
         ASK Vehicle TO SetParameters (Decoy, Count);
         ASK VehicleList TO Add (Vehicle);
      END FOR;
   END IF;
END METHOD;

ASK METHOD MakeBaseList (IN Num1 : INTEGER);
VAR
   Count : INTEGER;
BEGIN
   NumberBases := (Num1 DIV BaseFactor) + MinNumberBases;
   NEW(BaseList,1..NumberBases);
   FOR Count := 1 TO NumberBases
      NEW (BaseList[Count]);
      ASK SELF TO GetCoordinate (BaseList[Count]);
   END FOR;
END METHOD;

ASK METHOD PickBase (INOUT Location : PointType);
VAR
   Number : INTEGER;
BEGIN
   Number:= ASK RandomBase UniformInt (1, NumberBases);
   Location.XCoordinate := BaseList[Number].XCoordinate;
   Location.YCoordinate := BaseList[Number].YCoordinate;
END METHOD;

ASK METHOD GetCoordinate (INOUT Location : PointType);
BEGIN
   Location.XCoordinate := ASK XCoord UniformReal (MinX, MaxX);
   Location.YCoordinate := ASK YCoord UniformReal (MinY, MaxY);
END METHOD;

ASK METHOD PickSpeed (IN Mean: REAL) : REAL;
BEGIN
   RETURN (ASK RandomSpeed Triangular (Mean*.9, Mean, Mean*1.1));
END METHOD;

ASK METHOD DestroyTarget (IN Time : REAL; IN Target : VehicleObj);
BEGIN
   ASK Target TO Explode (Time);
   IF ASK Target ID <= NumberTELs
      INC(TargetsKilled);
   END IF;
   IF TargetsKilled = NumberTELs
      Hostile := FALSE;
      ASK System.CampaignTime TO GetSample (SimTime() - ASK System PreHostile);
      ASK System.NumberLaunches TO GetSample (FLOAT(Launch));
   END IF;
```

```
END METHOD;

ASK METHOD GenerateTargets;
VAR
Vehicle : VehicleObj;
   Location : PointType;
BEGIN
   FOREACH Vehicle IN VehicleList;
      IF ASK Vehicle Class = TEL
         NEW(Location);
         ASK SELF TO PickBase (Location);
         ASK Vehicle TO SetPosition (Location);
         DISPOSE (Location);
         TELL Vehicle TO ConductMission;
      ELSE
         NEW(Location);
         ASK SELF TO GetCoordinate (Location);
         ASK  Vehicle TO SetPosition (Location);
         ASK SELF TO GetCoordinate (Location);
         ASK  Vehicle TO SetDestination (Location);
         DISPOSE (Location);
         ASK  Vehicle TO ChangeVelocity (ASK SELF TO PickSpeed (AveSpeed));
         TELL Vehicle TO MoveTo (ASK Vehicle Destination);
      END IF;
   END FOREACH;
END METHOD;

ASK METHOD CountLaunch;
BEGIN
   INC(Launch);
END METHOD;

ASK METHOD ResetCampaign;
VAR
   Vehicle : VehicleObj;
   Count : INTEGER;
BEGIN
   Hostile := TRUE;
   TargetsKilled:=0;
   Launch:=0;
   FOREACH Vehicle IN VehicleList
      ASK Vehicle TO Reset;
   END FOREACH;
   FOR Count:=1 TO NumberBases
      ASK SELF TO GetCoordinate (BaseList[Count]);
   END FOR;
END METHOD;

ASK METHOD Reset;
VAR
   Vehicle : VehicleObj;
```

```
  Count : INTEGER;
BEGIN
  FOR Count := 1 TO ASK VehicleList numberIn
    Vehicle := ASK VehicleList TO Remove ();
    DISPOSE (Vehicle);
  END FOR;
  FOR Count :=1 TO NumberBases
    DISPOSE (BaseList[Count]);
  END FOR;
  DISPOSE (BaseList);
END METHOD;

END OBJECT;

END MODULE.
```

J. DBLUEFORCE.MOD

```
DEFINITION MODULE BlueForce;

FROM Moving IMPORT MovingObj, PointType, PathObj;
FROM GrpMod IMPORT QueueObj;
FROM OpArea IMPORT VehicleObj;
FROM RandMod IMPORT RandomObj;
FROM HomeBase IMPORT AircraftObj;

TYPE
  CommanderObj = OBJECT
    NumberAircraft, NumberSensors : INTEGER;
    ProbKill, MeanAttackVelocity, AttackRadius, OnStaTime,
    TurnAroundTime, Range, EstTELspeed : REAL;
    AttackValue, Direction, RemoteX, RemoteY,
    RandomSearchX, RandomSearchY : RandomObj;
    Squadron, Wave, TargetSet, RemoteQueue : QueueObj;
    CarrierPosit : PointType;
    ASK METHOD ObjInit;
    ASK METHOD ResetCampaign;
    ASK METHOD Reset;
    ASK METHOD ObjTerminate;
    ASK METHOD EchoInput;
    ASK METHOD SetParameters (IN Air, UGS : INTEGER);
    ASK METHOD GetCoordinate (INOUT Location : PointType);
    ASK METHOD Distance (IN P1, P2 : PointType; OUT D: REAL);
    ASK METHOD DetermineAssignment (IN Point : PointType);
    ASK METHOD AssignRandomSearch;
    ASK METHOD AssignTargets;
    ASK METHOD StartSearching;
    ASK METHOD DetermineDetections (IN Launcher : VehicleObj);
    ASK METHOD StopDetection (IN Number : INTEGER);
  END OBJECT;
```

```
  TargetObj = OBJECT (PathObj)
    {Time and Where}
    {GetPoint and PrintPoint}
    ASK METHOD GiveCoordinates (INOUT Location : PointType);
  END OBJECT;

  SensorObj = OBJECT
    SerialNumber : INTEGER;
    Location : PointType;
    Range : REAL;
    DetectionInfo : ARRAY INTEGER OF ACTID;
    ASK METHOD ObjInit;
    ASK METHOD Reset;
    ASK METHOD ObjTerminate;
    ASK METHOD SetParameters (IN Num1 : INTEGER; IN Num2 : REAL);
    ASK METHOD SetLocation (IN Loco : PointType);
    ASK METHOD SetDetection (IN Time : REAL; IN Number : INTEGER);
    TELL METHOD ReportDetection (IN Time: REAL);
    ASK METHOD StopDetection (IN Number : INTEGER);
  END OBJECT;

VAR
  SquadronCO : CommanderObj;

END MODULE.
```

K. IBLUEFORCE.MOD

```
IMPLEMENTATION MODULE BlueForce;

FROM OpArea IMPORT OpArea, VehicleObj;
FROM SimMod IMPORT SimTime, InterruptMethod;
FROM HomeBase IMPORT AircraftObj;
FROM Moving IMPORT PointType;
FROM System IMPORT System;
FROM GrpMod IMPORT QueueObj;
FROM MathMod IMPORT SQRT;
FROM RandMod IMPORT FetchSeed;

OBJECT CommanderObj;

ASK METHOD ObjInit;
BEGIN
  NEW (AttackValue);
  ASK AttackValue TO SetSeed (FetchSeed (7));
  NEW (Direction);
  ASK Direction TO SetSeed (FetchSeed (4));
  NEW (RemoteX);
  NEW (RemoteY);
  ASK RemoteX TO SetSeed (FetchSeed (2));
```

```
   ASK RemoteY TO SetSeed (FetchSeed (9));
   NEW (RandomSearchX);
   ASK RandomSearchX TO SetSeed (FetchSeed (5));
   NEW (RandomSearchY);
   ASK RandomSearchY TO SetSeed (FetchSeed (6));
   NEW (Squadron);
   NEW (TargetSet);
   NEW (CarrierPosit);
   NEW (Wave);
   NEW (RemoteQueue);
   INPUT (CarrierPosit.XCoordinate);
   INPUT (CarrierPosit.YCoordinate);
   INPUT (ProbKill);
   INPUT (MeanAttackVelocity);
   INPUT (AttackRadius);
   INPUT (OnStaTime);
   INPUT (TurnAroundTime);
   INPUT (EstTELspeed);
   INPUT (Range);
END METHOD;

ASK METHOD ResetCampaign;
VAR
   FA18 : AircraftObj;
   Remote : SensorObj;
   Target : TargetObj;
   Point : PointType;
   Count : INTEGER;
BEGIN
   FOR Count:= 1 TO ASK Wave numberIn
      ASK Squadron TO Add (ASK Wave TO Remove());
   END FOR;
   FOREACH FA18 IN Squadron
      ASK FA18 TO Reset;
   END FOREACH;
   FOREACH Remote IN RemoteQueue
      ASK Remote TO Reset;
   END FOREACH;
   FOR Count := 1 TO ASK TargetSet numberIn
      Target := ASK TargetSet TO Remove();
   DISPOSE (Target);
END FOR;
END METHOD;

ASK METHOD Reset;
VAR
   FA18 : AircraftObj;
   Remote : SensorObj;
   Target : TargetObj;
   Point : PointType;
   Count : INTEGER;
```

```
BEGIN
   FOR Count := 1 TO ASK Wave numberIn
      FA18 := ASK Wave TO Remove();
      DISPOSE (FA18);
   END FOR;
   FOR Count := 1 TO NumberAircraft
      FA18:=ASK Squadron TO Remove();
      DISPOSE (FA18);
   END FOR;
   FOR Count := 1 TO NumberSensors
      Remote := ASK RemoteQueue TO Remove();
      DISPOSE (Remote);
   END FOR;
   FOR Count := 1 TO ASK TargetSet numberIn
      Target := ASK TargetSet TO Remove();
      DISPOSE (Target);
   END FOR;
END METHOD;

ASK METHOD ObjTerminate;
VAR
   FA18 : AircraftObj;
   Remote : SensorObj;
   Target : TargetObj;
   Point : PointType;
   Count : INTEGER;
BEGIN
   DISPOSE (AttackValue);
    DISPOSE (Direction);
   DISPOSE (RemoteX);
   DISPOSE (RemoteY);
   DISPOSE (RandomSearchX);
   DISPOSE (RandomSearchY);
   DISPOSE (CarrierPosit);
   FOR Count := 1 TO ASK Wave numberIn
      FA18 := ASK Wave TO Remove();
      DISPOSE (FA18);
   END FOR;
   DISPOSE (Wave);
   FOR Count := 1 TO ASK Squadron numberIn
      FA18:=ASK Squadron TO Remove();
      DISPOSE (FA18);
   END FOR;
   DISPOSE (Squadron);
   FOR Count := 1 TO ASK RemoteQueue numberIn
      Remote := ASK RemoteQueue TO Remove();
      DISPOSE (Remote);
   END FOR;
   DISPOSE (RemoteQueue);
   FOR Count := 1 TO ASK TargetSet numberIn
      Target := ASK TargetSet TO Remove();
```

```
      DISPOSE (Target);
    END FOR;
    DISPOSE (TargetSet);
END METHOD;

ASK METHOD SetParameters(IN Air,UGS :INTEGER);
VAR
    FA18 : AircraftObj;
    Remote : SensorObj;
    Count, Total : INTEGER;
BEGIN
    NumberAircraft := Air;
    NumberSensors := UGS;
    FOR Count := 1 TO NumberAircraft;
      NEW (FA18);
      ASK FA18 TO SetParameters
        (Count,ProbKill,AttackRadius,OnStaTime,TurnAroundTime,CarrierPosit);
      ASK FA18 TO ChangeVelocity (MeanAttackVelocity);
      ASK FA18 TO SetPosition (CarrierPosit);
      ASK Squadron TO Add (FA18);
    END FOR;
    FOR Count := 1 TO NumberSensors
      NEW(Remote);
      ASK Remote TO SetParameters (Count,Range);
      ASK RemoteQueue TO Add (Remote);
    END FOR;
END METHOD;

ASK METHOD GetCoordinate (INOUT Location : PointType);
BEGIN
    Location.XCoordinate :=
    ASK RemoteX UniformReal (ASK OpArea MinX, ASK OpArea MaxX);
    Location.YCoordinate :=
    ASK RemoteY UniformReal (ASK OpArea MinY, ASK OpArea MaxY);
END METHOD;

ASK METHOD EchoInput;
BEGIN
    OUTPUT ();
    OUTPUT ("Prob of Kill = ", ProbKill);
    OUTPUT ("CV Location ", CarrierPosit.XCoordinate, " ", CarrierPosit.YCoordinate);
    OUTPUT ("Mean Velocity = ", MeanAttackVelocity);
    OUTPUT ("Attack Radius = ", AttackRadius);
    OUTPUT ();
    OUTPUT ("On Station Time = ", OnStaTime);
    OUTPUT ("Turn Around Time = ", TurnAroundTime);
    OUTPUT ();
    OUTPUT ("Estimated TEL speed = ", EstTELspeed);
    OUTPUT ();
    OUTPUT ("Number of Remote Sensors = ", NumberSensors);
    OUTPUT ("Range of Sensors = ", Range);
```

```
END METHOD;

ASK  METHOD Distance (IN P1, P2 : PointType; OUT D: REAL);
VAR
   X, Y : REAL;
BEGIN
   X := P1.XCoordinate - P2.XCoordinate;
   Y := P1.YCoordinate - P2.YCoordinate;
   D := SQRT(X*X + Y*Y);
END METHOD;

ASK METHOD DetermineAssignment (IN Point : PointType);
VAR
   D1, D2 : REAL;
   FA18 : AircraftObj;
   VirtualTarget: TargetObj;
BEGIN
   D1 := MAX (REAL);
   FOREACH FA18 IN Wave
      ASK SELF Distance (ASK FA18 Destination, Point, D2);
      D1 := MINOF(D1, D2);
   END FOREACH;
   ASK SELF Distance (Point, CarrierPosit, D2);
   IF D1 < D2 ;
      FOREACH FA18 IN Wave
         ASK SELF Distance (ASK FA18 Destination, Point, D2);
         IF D1 = D2;
            ASK FA18 TO ReceiveAssignment (Point);
         END IF;
      END FOREACH;
   ELSE
      NEW(VirtualTarget);
      ASK VirtualTarget TO GetPoint (SimTime(), Point);
      DISPOSE(Point);
      ASK TargetSet TO Add (VirtualTarget);
      ASK SELF TO AssignTargets;
   END IF;
END METHOD;

ASK METHOD AssignRandomSearch;
VAR
   Plane : AircraftObj;
   Point : PointType;
BEGIN
   WHILE (ASK Squadron numberIn > 0)
      Plane := ASK Squadron TO Remove();
      NEW(Point);
      Point.XCoordinate :=
      ASK RandomSearchX UniformReal (ASK OpArea MinX, ASK OpArea MaxX);
      Point.YCoordinate :=
      ASK RandomSearchY UniformReal (ASK OpArea MinY, ASK OpArea MaxY);
```

```
      ASK Plane TO SetDestination (Point);
      DISPOSE (Point);
      TELL Plane TO AttackTarget;
      INC (NumberSorties);
   END WHILE;
END METHOD;


ASK METHOD AssignTargets;
VAR
   Count : INTEGER;
   Plane : AircraftObj;
   VirtualTarget : TargetObj;
   Point : PointType;
   D , TravelTime, MaxR: REAL;
BEGIN
   IF ASK Squadron numberIn > 0 {Don't do anything if squadron empty}
      IF ASK TargetSet numberIn <> 0
      {***More Targets than aircraft***}
         IF ASK TargetSet numberIn > ASK Squadron numberIn
            WHILE (ASK Squadron numberIn > 0) AND (ASK TargetSet numberIn > 0)
               Plane := ASK Squadron TO Remove();
               VirtualTarget := ASK TargetSet TO Remove();
               NEW(Point);
               ASK VirtualTarget TO GiveCoordinates(Point);
               ASK SELF Distance (Point, CarrierPosit, D);
               TravelTime := (D / ASK Plane Velocity);
               MaxR := (SimTime() + TravelTime - ASK VirtualTarget Time)
                          * EstTELspeed;
               DISPOSE (VirtualTarget);
               IF MaxR > ASK Plane SearchRadius
                  ASK Squadron TO Add (Plane);
               ELSE
                  ASK Plane TO SetDestination(Point);
                  TELL Plane TO AttackTarget;
               END IF;
               DISPOSE(Point);
            END WHILE;
         ELSE
         {***More Aircraft than Targets***}
            WHILE (ASK TargetSet numberIn > 0) AND (ASK Squadron numberIn > 0)
               Plane := ASK Squadron TO Remove();
               VirtualTarget := ASK TargetSet TO Remove();
               NEW(Point);
               ASK VirtualTarget TO GiveCoordinates (Point);
               ASK SELF Distance (Point,CarrierPosit,D);
               TravelTime := (D / ASK Plane Velocity);
               MaxR := (SimTime() + TravelTime - ASK VirtualTarget Time)
                          * EstTELspeed;
               DISPOSE(VirtualTarget);
               IF MaxR > ASK Plane SearchRadius
```

```
                ASK Squadron TO Add (Plane);
              ELSE
                ASK Plane TO SetDestination (Point);
                TELL Plane TO AttackTarget;
              END IF;
              DISPOSE(Point);
            END WHILE;
          END IF;
          ASK SELF TO AssignRandomSearch; {for any leftovers}
        ELSE  {Executes if no targets in target set}
          ASK SELF TO AssignRandomSearch;
        END IF; {for targetset check}
      END IF; {for no aircraft}
    END METHOD;

    ASK METHOD StartSearching;
    VAR
      Remote : SensorObj;
      Location : PointType;
    BEGIN
      FOREACH Remote IN RemoteQueue
        NEW (Location);
        ASK SELF TO GetCoordinate (Location);
        ASK Remote TO SetLocation (Location);
        DISPOSE (Location);
      END FOREACH;
      {Airplanes begin to search regardless of number of UGS}
      ASK SELF TO AssignTargets;
    END METHOD;

    ASK METHOD DetermineDetections (IN Launcher : VehicleObj);
    VAR
      Xi, Xj, Vnorm, Vi, Vj, Ui, Uj, Xnorm, rad, td1, time : REAL;
      Remote : SensorObj;
    BEGIN
      FOREACH Remote IN RemoteQueue
        {Relative Space}
        Xi :=   ASK Launcher CurrentPosition.XCoordinate -
                ASK Remote Location.XCoordinate;
        Xj :=   (ASK Launcher CurrentPosition.YCoordinate) -
                (ASK Remote Location.YCoordinate);
        {Unit Vector for Target}
        Distance (ASK Launcher CurrentPosition, ASK Launcher Destination, Vnorm);
        Vi :=   ASK Launcher Destination.XCoordinate -
                ASK Launcher CurrentPosition.XCoordinate;
        Vj :=   ASK Launcher Destination.YCoordinate -
                ASK Launcher CurrentPosition.YCoordinate;
        Ui := Vi / Vnorm;
        Uj := Vj / Vnorm;
        {first test}
       Xnorm:=SQRT(Xi*Xi + Xj*Xj);
```

```
      IF Xnorm <= ASK Remote Range
         td1:=0.0;
         ASK Remote TO SetDetection (td1,  ASK Launcher SerialNumber);
      ELSE
         rad :=   (ASK Remote Range * ASK Remote Range) - (Xnorm * Xnorm) +
                  ((Ui*Xi + Uj*Xj)*(Ui*Xi + Uj*Xj));
         IF rad > 0.0
            td1:=( ((-Ui*Xi)+(-Uj*Xj)) - SQRT(rad) ) / ASK Launcher Velocity;
            IF td1 > 0.0
               ASK Remote TO SetDetection (td1,  ASK Launcher SerialNumber);
            END IF;
         END IF;
      END IF;
   END FOREACH;
END METHOD;

ASK METHOD StopDetection (IN Number : INTEGER);
VAR
   Remote : SensorObj;
BEGIN
   FOREACH Remote IN RemoteQueue
      ASK Remote TO StopDetection (Number);
   END FOREACH;
END METHOD;

END OBJECT;

OBJECT TargetObj;

ASK METHOD GiveCoordinates (INOUT Location : PointType);
BEGIN
   Location.XCoordinate:=Where.XCoordinate;
   Location.YCoordinate:=Where.YCoordinate;
END METHOD;

END OBJECT;

OBJECT SensorObj;
ASK METHOD ObjInit;
BEGIN
   NEW (Location);
   NEW    (DetectionInfo,
            1..(ASK OpArea NumberTELs + ASK OpArea NumberDecoys));
END METHOD;

ASK METHOD Reset;
VAR
   Count : INTEGER;
BEGIN
   FOR Count := 1 TO (ASK OpArea NumberTELs + ASK OpArea NumberDecoys);
      ASK SquadronCO TO StopDetection (Count);
```

```
  END FOR;
END METHOD;

ASK METHOD ObjTerminate;
BEGIN
  DISPOSE (Location);
  DISPOSE (DetectionInfo);
END METHOD;

ASK METHOD SetParameters (IN Num1 : INTEGER; IN Num2 : REAL);
BEGIN
  SerialNumber:=Num1;
  Range:=Num2;
END METHOD;

ASK METHOD SetLocation (IN Loco : PointType);
BEGIN
  Location.XCoordinate := Loco.XCoordinate;
  Location.YCoordinate := Loco.YCoordinate;
END METHOD;

ASK METHOD SetDetection (IN Time : REAL; IN Number : INTEGER);
BEGIN
  DetectionInfo[Number] := TELL SELF TO ReportDetection (Time);
END METHOD;

TELL METHOD ReportDetection (IN Time : REAL);
VAR
  Temp : PointType;
BEGIN
  WAIT DURATION Time;
    ON INTERRUPT
    TERMINATE;
  END WAIT;
  {*** UGS trigger aircraft ***}
  NEW(Temp);
  Temp.XCoordinate := Location.XCoordinate;
  Temp.YCoordinate := Location.YCoordinate;
  ASK SquadronCO TO DetermineAssignment (Temp);
END METHOD;

ASK METHOD StopDetection (IN Number : INTEGER);
BEGIN
  InterruptMethod (DetectionInfo[Number]);
END METHOD;
END OBJECT;
END MODULE.
```

L. DHOMEBASE.MOD

```
DEFINITION MODULE HomeBase;

FROM Moving IMPORT MovingObj, PointType;
FROM OpArea IMPORT VehicleObj;
FROM ListMod IMPORT QueueList;

TYPE
   AircraftObj = OBJECT (MovingObj);
   SerialNumber : INTEGER;
   CV : PointType;
   ProbKill, SearchRadius, OnStaTime, TurnAroundTime, PreviousAngle : REAL;
   GoToList :QueueList;
   ASK  METHOD SetParameters  (IN Number : INTEGER; IN Pk, R, St, Tt : REAL;
                                                IN Point:PointType);
   TELL METHOD FlyTo (IN There : PointType);
   ASK METHOD GetRandomDestination
      (IN Location : PointType; OUT Point :PointType);
   ASK METHOD Search : VehicleObj;
   ASK METHOD Attack (IN Target : VehicleObj; INOUT Continue : BOOLEAN);
   TELL METHOD AttackTarget;
   ASK METHOD ReceiveAssignment (IN Point : PointType);
   ASK METHOD ObjTerminate;
   ASK METHOD Reset;
   ASK METHOD TurnAround (IN WhichWall : STRING);
OVERRIDE
   ASK METHOD ObjInit;
END OBJECT;

END MODULE.
```

M. IHOMEBASE.MOD

```
IMPLEMENTATION MODULE HomeBase;

FROM Moving IMPORT PointType;
FROM BlueForce IMPORT SquadronCO;
FROM SimMod IMPORT SimTime;
FROM System IMPORT System;
FROM OpArea IMPORT OpArea, VehicleObj;
FROM MathMod IMPORT SQRT, COS, SIN, pi;
FROM RandMod IMPORT RandomObj;

OBJECT AircraftObj;

ASK METHOD ObjInit;
BEGIN
   NEW (CurrentPosition);
   NEW (Destination);
   NEW (CV);
   NEW (GoToList);
```

```
END METHOD;

ASK METHOD Reset;
VAR
 Count : INTEGER;
 Point : PointType;
BEGIN
FOR Count := 1 TO ASK GoToList numberIn
   Point:=ASK GoToList TO Remove();
   DISPOSE (Point);
END FOR;
   ASK SELF TO SetPosition (CV);
   ASK SELF TO SetDestination (CV);
END METHOD;

ASK METHOD ObjTerminate;
VAR
   Count : INTEGER;
   Point : PointType;
BEGIN
   DISPOSE (CurrentPosition);
   DISPOSE (Destination);
   DISPOSE (CV);
   FOR Count := 1 TO ASK GoToList numberIn
      Point := ASK GoToList TO Remove();
      DISPOSE (Point);
   END FOR;
   DISPOSE (GoToList);
 END METHOD;

ASK METHOD SetParameters  (IN Number : INTEGER; IN Pk, R, St, Tt : REAL;
                                          IN Point : PointType);
BEGIN
   SerialNumber := Number;
   ProbKill := Pk;
   CV.XCoordinate := Point.XCoordinate;
   CV.YCoordinate := Point.YCoordinate;
   SearchRadius := R;
   OnStaTime := St;
   TurnAroundTime := Tt;
END METHOD;

TELL METHOD FlyTo (IN There : PointType);
VAR
   D : REAL;
BEGIN
   StartTime := SimTime();
  ASK SELF TO SetDestination (There);
  ASK SELF Distance (CurrentPosition, Destination, D);
  WAIT DURATION  (D / Velocity);
  END WAIT;
```

```
   ASK SELF TO SetPosition (There);
END METHOD;

ASK METHOD GetRandomDestination
   (IN Location : PointType; OUT Point : PointType);
VAR
Theta, Upper, Lower, DeltaX, DeltaY : REAL;
BEGIN
   Upper := PreviousAngle - (pi / 2.0);
   Lower := PreviousAngle + (pi / 2.0);
   Theta := ASK SquadronCO.Direction UniformReal (Upper, Lower);
   PreviousAngle := Theta;
   WHILE PreviousAngle > 2.0 * pi;
      PreviousAngle := PreviousAngle - 2.0 * pi;
   END WHILE;
   DeltaX := 2.0 * SearchRadius * COS(Theta);
   DeltaY := 2.0 * SearchRadius * SIN(Theta);
   Point.XCoordinate := Location.XCoordinate + DeltaX;
   Point.XCoordinate := MINOF (ASK OpArea MaxX, Point.XCoordinate);
   Point.XCoordinate := MAXOF (ASK OpArea MinX, Point.XCoordinate);
   Point.YCoordinate := Location.YCoordinate + DeltaY;
   Point.YCoordinate := MINOF (ASK OpArea MaxY, Point.YCoordinate);
   Point.YCoordinate := MAXOF (ASK OpArea MinY, Point.YCoordinate);
   IF Point.XCoordinate = ASK OpArea MaxX
      ASK SELF TO TurnAround ("Right");
   END IF;
   IF Point.XCoordinate = ASK OpArea MinX
      ASK SELF TO TurnAround ("Left");
   END IF;
   IF Point.YCoordinate = ASK OpArea MaxY
      ASK SELF TO TurnAround ("Top");
   END IF;
   IF Point.YCoordinate = ASK OpArea MinY
      ASK SELF TO TurnAround ("Bottom");
   END IF;
END METHOD;

ASK METHOD Search : VehicleObj;
VAR
   Vehicle : VehicleObj;
   D : REAL;
   Location : PointType;
BEGIN
   FOREACH Vehicle IN OpArea.VehicleList
      IF ((NOT (ASK Vehicle Hidden)) AND (ASK Vehicle Alive))
         NEW (Location);
         ASK Vehicle TO CalcPosit (SimTime(), Location);
         ASK Vehicle Distance (ASK SELF CurrentPosition, Location, D);
         DISPOSE (Location);
         IF (D < ASK SELF SearchRadius)
            RETURN Vehicle;
```

```
      END IF;
    END IF;
  END FOREACH;
  RETURN NILOBJ;
END METHOD;

ASK METHOD Attack (IN Target : VehicleObj; INOUT Continue : BOOLEAN);
BEGIN
  IF Target <> NILOBJ
    IF ASK SquadronCO.AttackValue UniformReal (0.0, 1.0) < ASK SELF ProbKill
      ASK OpArea TO DestroyTarget (SimTime(), Target);
    END IF;
    Continue:=FALSE;  {Only one attack - hit or miss}
  ELSE
    Continue:=TRUE;
  END IF;
END METHOD;

TELL METHOD AttackTarget;
VAR
  BeginTime : REAL;
  Continue : BOOLEAN;
  NextPoint : PointType;
  Count : INTEGER;
BEGIN
  ASK SquadronCO.Wave TO Add(SELF);
  WAIT FOR SELF TO FlyTo(Destination);
  END WAIT;
  PreviousAngle:=0.0;
  BeginTime:=SimTime();
  Continue :=TRUE;
  ASK SELF TO Attack (ASK SELF TO Search, Continue);
  WHILE Continue;
    IF (SimTime() - BeginTime) < OnStaTime
      IF ASK GoToList numberIn > 0
        NextPoint := ASK GoToList TO Remove();
        ASK SELF TO SetDestination (NextPoint);
        DISPOSE (NextPoint);
      ELSE
        NEW (NextPoint);
        ASK SELF TO GetRandomDestination (CurrentPosition, NextPoint);
        ASK SELF TO SetDestination (NextPoint);
        DISPOSE (NextPoint);
      END IF;
      WAIT FOR SELF TO FlyTo (Destination);
      END WAIT;
      ASK SELF TO Attack (ASK SELF TO Search, Continue);
    ELSE
      Continue:=FALSE;
    END IF;
  END WHILE;
```

```
    ASK SquadronCO.Wave TO RemoveThis (SELF);
    ASK SELF TO SetDestination (CV);
    FOR Count := 1 TO ASK GoToList numberIn
      ASK SquadronCO TO DetermineAssignment (ASK GoToList TO Remove());
    END FOR;
    WAIT FOR SELF TO FlyTo(CV);
    END WAIT;
    WAIT DURATION TurnAroundTime;
    END WAIT;
    ASK SquadronCO.Squadron TO Add (SELF);
    IF ASK OpArea Hostile
      ASK SquadronCO TO AssignTargets;
    END IF;
END METHOD;

ASK METHOD ReceiveAssignment (IN Point : PointType);
BEGIN
    ASK GoToList TO Add (Point);
END METHOD;

ASK METHOD TurnAround (IN WhichWall : STRING);
BEGIN
    CASE WhichWall
      WHEN "Top" , "Bottom" :
        PreviousAngle := (2.0*pi) - PreviousAngle;
      OTHERWISE
        PreviousAngle := pi - PreviousAngle;
    END CASE;
END METHOD;
END OBJECT;
END MODULE.
```

LIST OF REFERENCES

Berhow, T., et. al., "A Theater Ballistic Missile (TBM) Counterforce Concept", Technical Report NPS-PH-94-006, Naval Postgraduate School, Monterey, CA, December 1993.

Cohen, Eliot A., Keaney, Thomas A., *Gulf War Air Power Survey, Summary Report*, U.S. Government Printing Office: Washington, D.C., 1993.

Dickson, Paul., *The Electronic Battlefield.* Bloomington and London: Indiana University Press, 1976.

Ehlers, M.A., *Countering the Third World Mobile Short Range Ballistic Missile Threat : An Integrated Approach*, Master's Thesis, Naval Postgraduate School, Monterey CA, September, 1992.

Fitzsimonds, James R. and Van Tol, Jan M., "Revolutions in Military Affairs", *Joint Forces Quarterly*, no.4, Spring 1994.

Gordon, Michael, *Admiral with High-Tech Dreams has Pentagon at War with Itself*, The New York Times, 12 December 1994.

Junker, V.L., *Tactical Unmanned Aerial Vehicles in a Proposed Joint Infrastructure to Counter Theater Ballistic Missiles*, Master's Thesis, Naval Postgraduate School, Monterey, CA, March, 1995.

Navias, Martin, *Ballistic Missile Proliferation in the Third World*, The International Institute for Strategic Studies, London, 1990.

Owens, W.A., "The Emerging System of Systems", U.S. Naval Institute *Proceedings*, vol. 121, no. 5, May 1995.

van Kan, Mark D., *A Probabilistic Target Classification and Description Model for Seismic Sensors*, Master's Thesis, Naval Postgraduate School, Monterey, CA, September, 1994.

INITIAL DISTRIBUTION LIST

		No. Copies
1.	Defense Technical Information Center 8725 John J. Kingman Rd., STE 0944 Ft. Belvoir, VA 22060-6218	2
2.	Dudley Knox Library Naval Postgraduate School 411 Dyer Rd. Monterey, CA 93943-5101	2
3.	Professor Donald P. Gaver, Code OR/Gv Naval Postgraduate School Monterey, California 93943-5000	2
4.	Professor Arnold H. Buss, Code OR/Bu Naval Postgraduate School Monterey, California 93943-5000	2
5.	Professor Wayne P. Hughes, Code OR/Hl Naval Postgraduate School Monterey, California 93943-5000	1
6.	Professor Michael G. Sovereign, Code OR/Sm Naval Postgraduate School Monterey, California 93943-5000	1
7.	Lieutenant Kenneth H. Munson, Jr. 3408 Douglas Drive Murrysville, Pennsylvania 15668	1